

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

INTERAKTÍVNÝ MODEL FITKITU

BAKALÁŘSKÁ PRÁCE

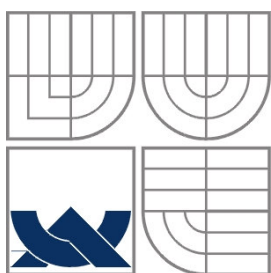
BACHELOR'S THESIS

AUTOR PRÁCE

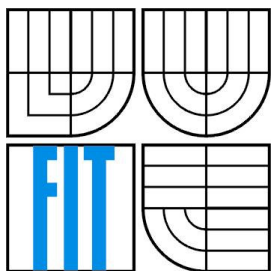
AUTHOR

MARTIN DANKO

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

INTERAKTIVNÍ MODEL FITKITU

INTERACTIVE FITKIT MODEL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN DANKO

VEDOUCÍ PRÁCE

SUPERVISOR

ING. ZDENĚK VAŠÍČEK

BRNO 2011

Abstrakt

Bakalářská práce se zabývá tvorbou interaktivního modelu platformy FITkit a možnostmi jeho prezentace v prostředí internetu. Přibližuje různé způsoby zobrazení trojrozměrného obsahu v internetových prohlížečích a přináší srovnání komerčních i bezplatných nástrojů. Podrobněji se věnuje technologii Flash, a v ní dostupným frameworkům umožňujícím manipulaci s 3D obsahem s využitím programovacího jazyka ActionScript. V druhé části se práce zabývá volbou vhodného frameworku, tvorbou modelu a implementací aplikace.

Abstract

The bachelor thesis deals with the creation of interactive model of FITkit platform and its presentation on the Internet. It shows different ways of presentation of the three-dimensional content in web browsers and provides a comparison of commercial as well as non-commercial tools. The introductory part is focused on the description of the Flash technology, the available 3D frameworks and the programming language ActionScript. The second part discusses the selection of the appropriate Flash framework and describes the implementation.

Klíčová slova

FITkit, Flash, Away3D, ActionScript, model, Flex SDK, interaktivní aplikace, tvorba 3D modelu

Keywords

FITkit, Flash, Away3D, ActionScript, model, Flex SDK, interactive application, 3D model creation

Citace

Danko Martin: Interaktivní model FITkitu, bakalářská práce, Brno, FIT VUT v Brně, 2011

Interaktívny model FITkitu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pána Ing. Zdeňka Vašíčka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Danko

18.05.2010

Poděkování

Za poskytnutí odbornou pomoc, cenné rady a připomienky počas konzultácií pri tvorbe praktickej ako aj teoretickej časti bakalárskej práce by som chcel poďakovať pánu Ing. Zdeňkovi Vašíčkovi.

© Martin Danko, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah	1
Úvod	2
1 Prezentácia 3D grafiky na internete	3
2 Technológia Adobe Flash	6
3 Podpora 3D vo Flashi	11
4 Návrh aplikácie	15
5 Voľba použitej technológie	17
6 Tvorba 3D modelu	21
7 Implementácia	26
8 Testovanie aplikácie	28
9 Možnosti rozšírenia aplikácie	31
10 Záver	32
Literatúra	33
Zoznam príloh	35

Úvod

FITkit je samostatná hardvérová platforma vyvíjaná ako učebná pomôcka pre študentov Fakulty informačných technológií v Brne (ďalej len FIT) , ktorej cieľom je podpora výučby hardvérových technológií a softvéru výpočtových systémov [1].

Na vývoji tejto platformy sa však postupom času, s pribúdajúcimi možnosťami využitia, ale i rastúcim okruhom priaznivcov nepodieľajú len zamestnanci a študenti FITu, ale aj externí spolupracovníci, a rady záujemcov o túto platformu z vonkajšieho prostredia mimo FIT naďalej rastú, preto je potrebné, aby aj verejnosť mohla byť informovaná o možnostiach využitia tejto platformy, a to nielen pri potrebách výučby.

Interaktívny model FITkitu je priamou okno reakciou na požiadavky verejnosti, ktorá tak bude mať možnosť dozvedieť sa o ňom niečo viac, a to nielen prostredníctvom textového popisu, či fotiek platformy, ale aj virtuálnym náhľadom na platformu FITkit v okne internetového prehliadača, ktorý je dnes dostupný nielen v PC, ale už takmer aj v každom mobilnom zariadení. Spoznávanie FITkitu sa tak otvorí celému svetu bez nutnosti fyzického kontaktu s platformou.

Práca je členená do 10 kapitol, pričom obsah jednotlivých kapitol je nasledujúci: prvá kapitola pojednáva o možnostiach prezentácie interaktívnej trojrozmernej grafiky na internete a to ako pomocou komerčných, tak aj nekomerčných nástrojov. V druhej kapitole si priblížime technológiu Flash od spoločnosti Adobe. Obsahom tretej kapitoly bude zoznámenie sa s podporou trojrozmernej grafiky vo Flashi. Štvrtá kapitola nás oboznámi s navrhovaným riešením vytvorenia interaktívnej aplikácie, ktorá by prezentovala model FITkitu. Porovnanie dostupných možností pre technológiu Flash, ako aj voľba tej najlepšej a bližšie oboznámenie sa s ňou bude obsahom piatej kapitoly. Šiesta kapitola popisuje vytváranie modelu FITkitu. Samotnej implementácii sa venuje siedma kapitola. Ôsma kapitola obsahuje vyhodnotenie navrhnutého prístupu z pohľadu rýchlosti. Predposledná kapitola diskutuje možné rozšírenia. Výsledky práce zhodnotíme v poslednej kapitole, kde si zároveň aj zhrnieme prínosy.

1 Prezentácia 3D grafiky na internete

Bolo len otázkou času, kedy sa začnú programátori dožadovať presadenia trojrozmernej (ďalej len 3D) grafiky aj na internete. V súčasnosti sú však možnosti jej, tzv. „on-line“ zobrazenia značne obmedzené, a pokiaľ chceme vytvárať interaktívnu aplikáciu v 3D na internete, musíme si vybrať spomedzi niekoľkých technológií. Z historického hľadiska bol prvou z možností formát VRML, neskôr spoločnosť Adobe priniesla podporu prezentácie 3D obsahu v PDF formáte, pre komerčný softvér od spoločnosti Autodesk - 3D Studio Max (tiež známy ako 3ds Max) vznikol nástroj TurnTool, no a v súčasnosti zažíva 3D grafika na poli internetu najväčší rozmach vďaka technológiám Adobe Flash a rýchle sa rozvíjajúcemu novému programovaciemu jazyku WebGL.

1.1 Komerčné nástroje

VRML (Virtual Reality Model Language) je grafický formát založený na deklaratívnom programovacom jazyku, ktorý bol navrhnutý najmä pre popis 3D scén, ktoré obsahujú aktívne aj pasívne objekty. Najbežnejším spôsobom zobrazovania, prehrávania alebo prehliadania VRML súborov, modelov a svetov je zásuvný modul pre podporu VRML, ktorý je známy ako VRML prehrávač, prehliadač, či ovládací prvok ActiveX. Existuje niekoľko týchto prehrávačov ako napr. Cosmo Player, Cortona 3D Viewer, Octaga Player, FreeWRL alebo OpenVRML. Niektoré z týchto modulov však fungujú len pod operačným systémom Microsoft Windows, prípadne sú vytvorené len pre konkrétny internetový prehliadač. Rozšírením formátu VRML je formát X3D, ktorý navyše prináša podporu XML syntaxe. Veľké množstvo prehrávačov formátu VRML je kompatibilných aj s formátom X3D a naopak. [2 a 3]

Formát VRML vzbudil veľký záujem, ale jeho využitie nikdy nenašlo širšie uplatnenie. Jedným z dôvodov tejto skutočnosti mohla byť aj nedostatočná dostupnosť širokopásmového internetu. V čase, keď sa formát VRML stával populárnym väčšina používateľov, ako firemných, tak aj osobných, bola k internetu pripojená prostredníctvom pomalej pevnej linky. To malo nešťastný postranný následok, že používatelia museli pomerne dlhšiu chvíľu čakať len na to, aby videli malú, zle osvetlenú štvorcovú miestnosť, v ktorej skreslený obsah „visel“ na zdánlivo náhodných miestach. S VRML sa experimentovalo prevažne v oblasti výskumu a vzdelávania, kde je otvorená špecifikácia jazyka tým najcennejším. VRML formát však našiel svoje uplatnenie najmä v CAD systémoch, kde slúži na prenášanie a výmenu 3D modelov.

Adobe Acrobat ponúkol vývojárom komerčnú sadu pre vývoj softvéru s možnosťou jednoduchej integrácie 3D modelov do formátu PDF súborov, ktoré sú obklopené 2D obsahom. Tejto možnosti bolo dosiahnuté za použitia JavaScriptu, ktorý je priamo podporovaný PDF formátom, ale Acrobat prišiel aj so špecializovaným rozhraním pre programovanie aplikácií, ktoré rovnako môže byť použité na prehrávanie animácií, ovládanie, nastavenie farieb, viditeľnosti a širokú škálu ďalších aspektov 3D modelu priamo z hlavného dokumentu v PDF formáte. K prehrávaniu interaktívneho 3D obsahu v PDF súboroch je však nevyhnutné tieto súbory otvárať práve, a jedine len v aplikácii Adobe Acrobat Reader (podpora 3D je od verzie 7 a vyššej). [4]

TurnTool je z technického hľadiska doplnkom, ktorý pridáva do komerčného softvéru 3D Studio Max možnosť exportovania do špeciálneho formátu TNT súboru. Objekty, animácie, či celé scény vrátane textúr je tak možné prezerať v internetových prehliadačoch Internet Explorer a Mozilla Firefox pomocou ďalšieho zásuvného modulu s názvom TurnTool 3D Viewer. Existuje bezplatná verzia nástroja TurnTool, ktorá vo výslednej scéne zobrazuje logo spoločnosti s odkazom alebo platená verzia, ktorá už žiadnu „reklamu“ neobsahuje. [5]

1.2 Bezplatné nástroje

Medzi najväčších súperov ponúkajúcich prezentáciu 3D grafiky na internete v bezplatnej sfére sa radia konkurujúci si Adobe Flash a WebGL. [6]

Na jednej strane je tu výhoda rozhrania WebGL, ktorého princípom je využitie JavaScriptu spúšťaného priamo v okne prehliadača, čo znamená, že môže využívať všetky prostriedky dostupné internetovému prehliadaču, vrátane hardvérovej podpory grafickej karty pomocou OpenGL. Keďže WebGL bude spoliehať len na JavaScript, nebude potrebný žiadny dodatočný softvér na strane prehliadača, a k behu aplikácie vytvorenej pomocou WebGL je potrebná len podpora skriptovacieho jazyka a HTML 5 elementu „Canvas“, čo by malo zabrániť strate kompatibility. [7]

Problémom však v súčasnosti je veľmi malá podpora WebGL zo strany prehliadačov, pretože okrem Google Chrome (od verzie 9) ho podporujú prevažne len testovacie verzie ostatných prehliadačov, a firma Microsoft zatiaľ podporu vo svojom prehliadači Internet Explorer (IE) úplne ignoruje. Organizácia Khronos Group, ktorá stojí za vývojom OpenGL totiž ešte stále nepublikovala oficiálnu špecifikáciu štandardu. Bez podpory IE je budúcnosť WebGL značne obmedzená, pretože Microsoft má príliš veľký vplyv na vývoj 3D grafiky na internete, čo je dôsledkom masového rozšírenia prehliadača IE. V minulosti Microsoft podporu WebGL zamietol, ale zvažuje sa jeho podpora v novo uvedenej verzii prehliadača IE9, ktorý už disponuje aj hardvérovou akceleráciou vykresľovania. Ani pri uvedení výslednej verzie IE9, však podpora implementovaná nebola, takže implementácia WebGL v prehliadači IE ostáva naďalej nezodpovedanou otázkou.

Na druhej strane, pre zobrazenie Flash-ového obsahu je potrebná inštalácia zásuvného modulu. Tento modul existuje pre všetky súčasné operačné systémy, ako aj internetové prehliadače, a široké pole pôsobnosti Flashu je jeho hlavnou dominantou. Spoločnosť Adobe stojaca za vývojom Flash technológie uvádza dostupnosť na 98-99% osobných počítačov súčasnosti. V nových verziách (10 a vyššie), je už čiastočne zahrnutá aj podpora hardvérovej akcelerácie, ktorej plné nasadenie s využitím 3D grafiky je však odkladané, pretože spoločnosť Adobe sa zamerala na ciele s vyššou prioritou, akými sú: podpora mobilných zariadení, tabletov, televízorov, či zobrazovanie na viacerých obrazovkách. Tento fakt, ale neznehodnocuje kvality Flash technológie, pretože zavedenie podpory pre GPU je plánové na rovnaké obdobie ako uvedenie oficiálnej verzie WebGL. Zatiaľ spoločnosť Adobe uvoľnila beta verziu knižnice s plnohodnotnou podporou 3D akcelerácie, ktorá je dostupná pod kódovým označením Molehill. Táto knižnica dokáže pre renderovanie využívať ako OpenGL, tak aj rozhranie DirectX, a vo finálnej verzii nepochybne dokáže podať vysokú kvalitu renderovania 3D grafiky pri využití minima systémových prostriedkov. [8]

	<i>Hardvérová akcelerácia</i>	<i>Vyžadovaný zásuvný modul</i>	<i>Súčasná podpora 3D</i>	<i>Verzia k dispozícii</i>	<i>Podpora mobilných zariadení</i>	<i>Podpora prehliadačov</i>
FLASH	čiastočne	áno	vo vývoji	stabilná	áno	všetky
WebGL	áno	nie	vo vývoji	nestabilná	nie	testovacie verzie, Chrome, okrem IE

Tabuľka 1.1 – Zhrnutie výhod a nevýhod jednotlivých bezplatných technológií

V tabuľke 1.1 vidíme krátke zhrnutie výhod a nevýhod jednotlivých technológií, z ktorej vyplýva, že najväčšou výhodou WebGL je práve fakt, že k jeho fungovaniu nie je vyžadovaný žiadny zásuvný, pretože implementácia tohto rozhrania je závislá len od tvorcov internetových prehliadačov. V čom ale WebGL technológii Flash zatiaľ nemôže konkurovať je okrem podpory mobilných zariadení práve to najdôležitejšie – dostupnosť.

WebGL pravdepodobne nájde svoje miesto v internetovom 3D svete budúcnosti, ale v súčasnosti patrí prvotné a jediné miesto v tomto poli technológii Flash, ktorá neohrozene vládne naprieč všetkým operačným systémom a prehliadačom. Flash však okrem natívnej podpory 3D grafiky, ponúka množstvo rozšírení vo forme frameworkov (framework – sada tried a knižníc, ktorá rozširuje funkcionality pôvodného softvéru.), ktoré aj do súčasných verzií Flashu dokážu pridať plnú podporu pre prácu s 3D grafikou.

2 Technológia Adobe Flash

Položme si základnú otázku: „Čo je to Flash?“. Flash by sa dal v skratke charakterizovať ako „veľa vecí pre veľa ľudí“, teda univerzálna multimediálna platforma. Za dobu, čo technológia Flash existuje v počítačovom svete už bola nástrojom pre tvorbu animácií, tvorbu webových stránok, platformou pre vývoj hier, on-line prehrávanie videí, ale v neposlednom rade aj nástrojom pre vývoj aplikácií. V dobe súčasného internetu je Flash označovaný ako nástroj RIA – „Rich Internet Applications“, čo v preklade znamená: nástroj bohatých internetových aplikácií. Všetok obsah vytvorený pomocou technológie Flash môže byť zobrazený na rôznych počítačových systémoch a zariadeniach prostredníctvom prehrávača Adobe Flash Player, ktorý je k dispozícii zadarmo. [9]

Dôkazom toho, že Flash je najvhodnejšou voľbou pre tvorbu interaktívneho obsahu na poli internetu svedčia aj štatistiky [10], ktoré hovoria, že až 85% najnavštevovanejších internetových stránok používa k zobrazeniu celého alebo časti obsahu Flash, 75% videí prezretých na internete je prehrávaných prostredníctvom aplikácie Flash Player, 98% podnikov sa pri tvorbe internetových stránok spolieha na Flash Player, a 70% on-line hier je vytvorených práve vo Flashi.

2.1 Flash - výhody a nevýhody

K rozšíreniu Flashu vo všetkých smeroch, ale najmä na internete, pomohla malá veľkosť výsledných súborov, pretože sa ukladajú vo vektorovom formáte (Flash ale dokáže pracovať ako s vektorovou, tak aj s rastrovou grafikou). Dokonca aj zložitejšie animácie, hry a aplikácie majú vo výsledku veľkosť len desiatok, či stoviek kilobajtov, čo zároveň s rastúcou rýchlosťou internetu prináša možnosť interaktívneho obsahu, ktorý je dostupný takmer okamžite. Flash má taktiež implementovaný aj vlastný programovací jazyk ActionScript, ktorý slúži k rozvoju interaktívnych animácií a vývoju robustných aplikácií.

Ďalšou výhodou Flashu je prehrávanie obsahu priamo v okne internetového prehliadača bez nutnosti sťahovania alebo inštalovania. Podporuje obojsmerný prenos zvuku a videa, dokáže zachytávať aj používateľské vstupy pomocou počítačovej myšky, klávesnice, mikrofónu a kamery.

Zatiaľ čo väčšina používateľov pociťuje skúsenosť s Flashom ako obohacujúcu, široké možnosti využitia priniesli aj tienisté stránky, čo malo negatívny dopad, ktorý sa prejavil najmä v oblasti reklamy, ktorá sa často javila ako rušivá alebo priam obťažujúca. Preto začali vznikať aj nástroje, ktoré dovoľujú Flashový obsah zablokovať. Flash bol preto kritizovaný na nepriaznivý vplyv na použiteľnosť internetových stránok.

Na druhej strane Flash drží krok aj s vývojom ostatných technológií, dôkazom toho je aj natívna podpora 64-bitových systémov, hardvérovo akcelerované vykresľovanie grafiky, ktoré prináša zrýchlenie grafického výkonu, a menšie zaťaženie procesora, tým, že prenáša výpočty z CPU na GPU. Toto bola reakcia na výzvy používateľov, ktorí sa sťažovali na pomerne vysoké zaťaženie procesora pri prehrávaní flashového obsahu. Flash dokáže taktiež vykresľovať grafiku v sub-pixeloch, čo má za následok ostrejší a príjemnejší vzhľad zobrazovanej grafiky, zatiaľ čo pri iných spôsoboch vykresľovania môže dochádzať k rozporom a nespoľahlivosti pri vykresľovaní. [11].

2.2 Adobe Flash Player

„Softvér Adobe® Flash® Player je aplikačný modul runtime pre viaceré platformy založený na prehliadači, ktorý umožňuje plynulé zobrazenie pútavých aplikácií, obsahu a videí na rôznych

obrazovkách a v rôznych prehliadačoch. Softvér Flash Player 10.2 umožňuje zobrazenie nádherného videa s vysokým rozlíšením (HD), rýchlejšie vykresľovanie grafiky a dosiahnutie vysokého výkonu na obrazovkách mobilných zariadení. Je navrhnutý tak, aby využíval natívne schopnosti zariadení, čím používateľom prináša bohatšie a intenzívnejšie zážitky.“ [12].

Na oficiálnych stránkach [12] prehrávača je možné pozrieť si aj kompletný zoznam podporovaných operačných systémov ako aj internetových prehliadačov, z ktorého je možné vyčítať, že možnosti využitia tohto prehrávača sú takmer neobmedzené, a žiadny systém súčasnej doby nebude mať s prehrávaním obsahu najmenší problém. Adobe Flash Player podporuje skriptovací jazyk ActionScript vo všetkých jeho verziách. V súčasnosti je dostupná už aj jeho verzia pre mobilné zariadenia, tablety a niekoľko ďalších elektronických zariadení, pre ktoré je určená špeciálna verzia Flash Lite. Pri tvorbe aplikácie bola k prehrávaniu používaná verzia Flash Player 10.2, pričom aktuálna verzia je vždy k stiahnutiu na oficiálnych stránkach prehrávača [13].

2.3 ActionScript

2.3.1 História a vývoj jazyka ActionScript

ActionScript (ďalej len AS) je objektovo-orientovaný jazyk pôvodne vyvinutý firmou Macromedia Inc., ktorú v súčasnosti vlastní Adobe Systems. AS je dialektom ECMAScript, čo znamená, že má rovnakú syntax a sémantiku ako široko známy skriptovací jazyk JavaScript. Je používaný primárne pre tvorbu webových stránok a softvéru so zameraním na platformu Adobe Flash Player, pričom jeho použitie na webových stránkach je umožnené vložением spustiteľného SWF súboru (ShockWave Flash), ktorý sa vkladá ako objekt alebo môže byť spustený priamo v okne prehrávača. Jazyk ako taký má otvorené jadro, o čom svedčí aj fakt, že jeho špecifikácia je ponúkaná bezplatne a s otvoreným jadrom je šírený aj kompilátor tohto jazyka, ktorý je súčasťou balíka Adobe Flex.

ActionScript bol pôvodne navrhnutý pre ovládanie jednoduchých animácií dvoj-rozmerných vektorov vytvorených v Adobe Flash (predtým Macromedia Flash). Spočiatku sa zameriaval len na animácie, v skorých verziách ponúkal len niekoľko interaktívnych funkcií pre prácu s flashovým obsahom, a jeho skriptovacie možnosti boli veľmi obmedzené. Neskoršie verzie už pridávali nové funkcie, ktoré vývojárom umožňovali tvorbu on-line hier a bohatých internetových aplikácií, vrátane streamovaných médií (ako napríklad video a zvuk). V súčasnosti už AS ponúka možnosti, ktorých využitie sa nájde aj v oblastiach ako sú databázové systémy, či jednoduchá robotika.

S príchodom Macromedia Flash MX 2004 nastúpil AS už vo verzii 2.0 – skriptovací programovací jazyk lepšie prispôsobený pre vývoj flashových aplikácií. Často je totiž možné ušetriť množstvo času naskriptovaním niečoho ako vytváraním animácie, čo zvyčajne prináša taktiež vyššiu mieru flexibility pri úpravách.

Od nástupu Flash Playeru vo verzii 9 (rok 2006) bola uvoľnená aj najnovšia verzia jazyka ActionScript 3.0. AS 3.0 je objektovo-orientovaný programovací jazyk, ktorý poskytuje ďaleko väčšiu kontrolu a znovupoužitie kódu pri tvorbe komplexných flashových aplikácií. Táto verzia jazyka je prioritne kompilovaná a spúšťaná na novej verzii ActionScript Virtual Machine (označovanej ako AVM2). Z tohto dôvodu, kód napísaný v jazyku AS 3.0 je všeobecne zameraný na Flash Player vo verzii 9 a vyššej, čiže nepracuje s predchádzajúcimi verziami. Avšak v rovnakej dobe pracuje jazyk AS 3.0 až desaťkrát rýchlejšie ako ostatné verzie jazyka AS. [14]

2.3.2 Priblíženie jazyka ActionScript 3.0

```
package com.example {
    import flash.text.TextField;
    import flash.display.Sprite;

    public class Greeter extends Sprite {
        public function Greeter(){
            var txtHello: TextField = new TextField();
            txtHello.text = "Hello World";
            addChild(txtHello);
        }
    }
}
```

Ukážka 2.1 - Program „Hello world“ v jazyku ActionScript 3.0

Ako vidíme vo vyššie uvedenom príklade (ukážka 2.1), nová verzia jazyka prináša vyššiu úroveň abstrakcie programovacieho jazyka od integrovaného flashového prostredia. Pri zložitejších a komplexnejších programoch, sa však tento prístup ukazuje ako omnoho výhodnejší a prehľadnejší.

ActionScript sa primárne skladá z elementárnych a základných dátových typov, z ktorých môžu byť odvodené nové dátové typy. Dátové typy jazyka AS sa najviac podobajú dátovým typom jazyka Java. AS 3.0 bol však kompletne prebudovaný, a preto sú dátové typy ako aj ich dedičnosť kompletne rozdielne. Kým AS 2.0 ponúkal 4 dátové typy najvyššej úrovne (String, Number, Boolean, Object), tak AS 3.0 už ponúka 7 dátových typov najvyššej úrovne (Boolean, Int, Null, Number, String, Uint, Void), podobné rozdiely by sme našli aj u komplexných dátových typov.

2.3.3 Ochrana obsahu

ActionScript prináša taktiež ochranu zdrojového kódu, pretože množstvo vývojárov tvoriacich vo Flashi, ktorý využívajú jeho výhody pri tvorbe animácií a interaktívnych aplikácií, či webových stránok, nechce vystavovať svoj zdrojový kód celému svetu. Avšak ako u všetkých pokročilých jazykov, skompilovaný kód je uložený v lokálnom SWF súbore, ktorý je možné dekompilovať späť do zdrojového kódu a jeho zdrojov. Aj keď sú už niektoré dekompilátory schopné takmer kompletnej rekonštrukcie pôvodných zdrojových súborov až na kódový jazyk použitý pri tvorbe aplikácie, výsledky sa prípad od prípadu líšia.

Do opozície voči týmto dekompilátorom sa stavajú obfuskátory (alebo tiež zatemňovače) jazyka ActionScript, aby predišli kradnutiu zdrojových kódov, čím boli často porušované aj autorské práva. Rastúca kvalita obfuskátorov prináša implementovanie lexikálnych transformácií, ako sú napríklad: premenovanie identifikátorov, kontrola toku transformácie, a transformácia abstrakcie dát. To všetko sťažuje dekompiláciu výsledných SWF súborov, čo robí výstup vygenerovaný dekompilátorom takmer nečitateľným, a teda aj nepoužiteľným. Jednoduchšie obfuskátory do kódu naopak vkladajú tzv. pasce pre dekompilátory, ktoré ich funkciu komplikujú.

Flash ako jediný ponúka aj podporu DRM (viac viď [15]), ktorá zastrešuje technické metódy pre ochranu digitálnych médií (hudba, filmy, hry, obrazové umenie, a pod.).

2.4 Open Source Flex SDK

Ak nepoužívame program Adobe Flash, ktorého licenciu je potrebné si zakúpiť, existuje bezplatná varianta, ktorú budeme potrebovať, aby sme mohli využívať knižnice, ktoré obsahujú flashové komponenty, na to slúži práve vývojový framework Adobe Flex SDK, pre naše účely je vhodnejšia jeho odnož s otvoreným jadrom - Open Source Flex SDK [16]. Tento vysoko produktívny a bezplatný framework, ktorý je šírený s otvoreným jadrom nám poskytuje všetky možnosti pre vytváranie webových expresívnych aplikácií, ktoré nie sú závislé na použitej platforme operačného systému ani internetovom prehliadači, vďaka využitiu Adobe Flash Player a Adobe AIR. Poskytuje nám programovací model, ktorý podporuje množstvo návrhových vzorov pre vývojárov.

Kým Flex aplikácia môže byť vytvorená s využitím Open Source Flex SDK, softvér Adobe Flash Builder môže urýchliť vývoj týchto aplikácií prostredníctvom inteligentných funkcií pre podporu kódovania, interaktívneho krokového ladenia, či vizuálneho návrhu vzhľadu používateľského rozhrania. Vývojári môžu vytvárať intuitívnejšie aplikácie, ktoré umožňujú zákazníkom, partnerom aj zamestnancom spravovať a využívať údaje (dáta) efektívnejšie. Výsledkom čoho je zvýšená produktivita a väčšia spokojnosť používateľov.

Okrem toho Flex prináša moderný programovací model, ktorý prináša podporu bežných návrhových vzorov. Flex využíva jazyk MXML, deklaratívny jazyk založený na XML, a objektovo-orientovaný ActionScript 3.0. Flex obsahuje aj bohatú knižnicu komponentov s veľkým množstvom osvedčených rozšírení, ktoré prinášajú obohatenie používateľského rozhrania pre vytváranie ešte bohatších aplikácií, ale aj interaktívny debugger pre ladenie Flex aplikácií [17].

Flex aplikácia na rozdiel od stránkovo založenej HTML aplikácie slúži ako prezentačná vrstva medzi používateľom a prehliadačom, čo v praxi znamená, že pri nejakej zmene v aplikácii Flex, nie je potrebné, aby sa internetová stránka znovu načítala. Flex v spolupráci s Flash Playerom poskytujú mnoho užitočných spôsobov, ako odosielať a načítavať dáta do, a zo strany serverových komponentov bez nutnosti opätovného načítania. Toto riešenie prinieslo množstvo výhod oproti v minulosti využívanému HTML s využitím jazyka JavaScript, pretože asynchrónne načítania dát sa tak stalo bežnou praxou vo vývoji aplikácií založených na HTML.

K behu aplikácie vytvorenej vo frameworku Flex SDK je potrebný len prehrávač Adobe Flash Player. Rôzne verzie Open Source Flex SDK je možné stiahnuť z oficiálnych stránok frameworku [18].

2.5 Vývoj aplikácií

2.5.1 FlashDevelop

FlashDevelop je bezplatný editor zdrojového kódu s otvoreným jadrom. Jeho používanie je jednoduché, a naučiť sa s ním pracovať je vskutku otázkou niekoľkých minút. Ponúka všetky možnosti, ktoré by sme mohli od editora požadovať, vrátane podpory šablón a farebného zvýrazňovania syntaxe, pričom sa zameriava aj na jeden z pre nás kľúčových programovacích jazykov ActionScript, presnejšie na verzie tohto jazyka s číslom 2 a 3. Prostredie editora je intuitívne a flexibilné, takže je možné prispôbiť ho požiadavkám každého programátora. Samozrejmosťou je aj simultánne otvorenie viacerých súborov, medzi ktorými môžeme prepínať pomocou kariet. Editor podporuje aj vytváranie vlastných zásuvných modulov, čo sme využili, a obohatili ho tak, aby spolupracoval s nami používaným 3D engine-om. Veľmi užitočná je taktiež funkcia dokončovania kľúčových slov (IntelliSense) a názvov premenných, ktorá nám neraz pomôže ušetriť množstvo času,

ktorý by sme inak strávili vypisovaním komplexných názvov premenných a ich atribút. Nastavenie editora pre kompiláciu s využitím flashového kompilátora je rovnako jednoduché ako navigácia v zdrojovom kóde. Integrovaný manažér projektov nám zasa pomáha udržať si prehľad vo všetkých súboroch prislúchajúcich k nášmu projektu, vrátane ich tried a vlastností, ktoré zobrazuje usporiadané v stromovej štruktúre. Pri ladení programu využijeme funkcie integrovaného debuggera jazyka ActionScript. FlashDevelop je možné stiahnuť z oficiálnych stránok programu [19].

3 Podpora 3D vo Flashi

Prvá vec, ktorú by sme mali o Flashi vedieť je, že technológia Flash sama o sebe 3D zobrazovanie zatiaľ nepodporuje. Niektoré webovo-orientované nástroje ako napríklad Director, Anark alebo Cult3D podporujú 3D zobrazovanie natívne, čím uľahčujú začlenenie 3D prvkov do internetových stránok. Čo však Flash vie, je ako správne zobrazovať vektorové útvary na obrazovke a ako vypočítat matematické výrazy. To umožňuje vývojárom vytvoriť falošnú ilúziu 3D obsahu. Všeobecne sa všetko točí okolo 2 spôsobov vytvárania tejto ilúzie. Jedným z nich je, že sa vytvorí reálna 3D scéna v programe mimo prostredie Flash, ktorá je pred-vyrenderovaná a následne vložená do flashového obsahu, ktorý sa prehráva snímok po snímku podobne ako video. Druhým spôsobom je dynamické vytvorenie matematicky vypočítavaného 3D zobrazenia od nuly cez ActionScript.

Kapitola a jej podkapitoly čerpajú z [20].

3.1 Metódy interpretácie tretieho rozmeru (3D)

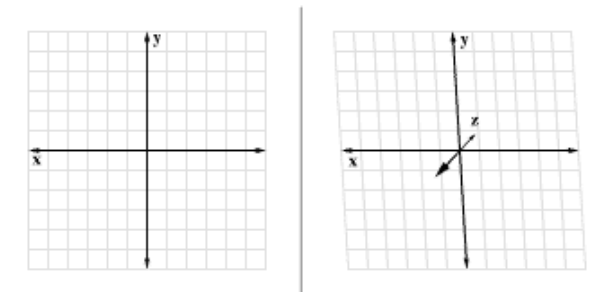
V prípade, že by sme zvolili cestu pred-vyrenderovanej scény, môžeme využiť nástroje ako sú Swift3D alebo Plasma. Čo tieto programy robia je, že vyrenderujú 3D scénu, ktorú následne vyexportujú do SWF súboru, ktorý je formátom umožňujúcim prehrávanie akéhosi videa, ktoré je sekvenciou snímok nasledujúcich za sebou. Takto vytvorené súbory môžu byť potom vložené do Flashu, ktorý s nimi dokáže manipulovať. Tieto produkty odvedú všetku náročnú a tzv. „špinavú“ prácu s 3D animáciami a zobrazovaním scén, čím značne redukovávajú požiadavky na Flash samotný, ktorý v týchto prípadoch slúži len ako prehrávač snímok animácie. Tieto produkty sú preto skvelým riešením pre 3D animácie, ktoré sú prehrávané v slučkách alebo prehrávanie videí. Ale vzhľadom k ich linearite je značne obmedzená možnosť interakcie. Inými slovami, je veľmi málo možností, čo môžeme robiť s týmto 3D obsahom, keď sa začne prehrávať. Všetko je pred-vyrenderované dopredu, tým pádom nič nemôže byť zmenené vo chvíli, keď sa už obsah prehráva vo Flash Playeri.

Výhodou tejto metódy je fakt, že akýkoľvek druh 3D vykresľovania a 3D výpočtov je postavený len na matematických výpočtoch, ktoré Flash zvláda bravúrne. To znamená, že sa môžeme plne sústrediť na prácu už od začiatku, a použiť ActionScript k vytvoreniu základov 3D scény s využitím základnej matematiky, na ktorej je 3D zobrazovanie založené. Následne môžeme vďaka týmto výpočtom a kresliacemu rozhraniu Flashu vytvárať vlastné vektorové tvary k vyvolaniu dojmu, že na scéne sú skutočné 3D objekty. Flash samotný však nevie o tom, že niečo z toho sa odohráva v 3D. Ten vidí len matematické výpočty, kreslenie a pohyby na obrazovke. To, čo vidí používateľ na druhej strane je úžasný obraz technickej zdatnosti programátora, a vynikajúce, aj keď vcelku jednoduché, stvárnenie poskytované formou 3D.

Existuje viacero stupňov 3D zobrazenia, ktoré môžeme implementovať vo Flashi. Môžeme začať od jednoduchého približovania a oddiaľovania obrazu až k plne dynamickej rotácii 3D objektu. Hlavnou nevýhodou tejto možnosti je však fakt, že všetko musíme vytvoriť z ničoho.

Rozdiel medzi reálnym 3D a normálnym (statickým) pohľadom sa vzťahuje na jeden najdôležitejší pojem, a tým je zmena mierky perspektívy. Je to myšlienka umiestnenia a rozsahu útvaru na obrazovke, ktorý je schopný reagovať na základe svojho teoretického umiestnenia (vzdialenosti) od obrazovky samotnej, čím vzniká aspekt 3D polohy. S normálnym 2D umiestnením na obrazovke, z ktorého Flash vychádza je vzdialenosť od obrazovky vždy rovnaká, inak by sme mohli tiež povedať, že útvar je vždy priamo v obraze. S tým súvisí aj smer pohybu čohokoľvek v tomto „plytkom“ priestore, na základe ktorého je pohyb obmedzený len na pozíciu vľavo/vpravo

pozdĺž vodorovnej X-ovej osi alebo hore/dole pozdĺž zvislej Y-ovej osi. Pohyb dopredu/dozadu pozdĺž pomyselnéj osi Z je v tomto prípade možný, aj keď v normálnom 2D zobrazení je neviditeľný. Situácie bližšie popisuje obrázok 3.1.



Obrázok 3.1 – Osa Z, ktorú v 2D zobrazení nevidíme, sa „vráti“ do priestoru [20]

Technicky teda osa Z vo Flashi ani v skutočnosti neexistuje. Flash sám teda naďalej rozoznáva len osi X a Y. Je len na programátorovi, aby túto osu Z vytvoril a naučil Flash rozoznávať ju ako keby tam naozaj bola. Osa Z je teda potrebná tam, kde je využívaná vzdialenosť, čiže zmena mierky perspektívy.

Teraz už síce vieme akým spôsobom sa dá prezentovať tretí rozmer – osa Z v 2D zobrazení, ale nemáme reálne prostriedky pre detekciu pohybu v tomto rozmere. Aby sme odlíšili tento pohyb, budeme musieť prijať isté opatrenia, aby bolo zjavné, že sa opierame o tieto vizuálne zachytne body. Dve hlavné techniky, ktoré dokážu tieto rozdiely správne interpretovať sú zmena rozsahu a prekrývanie.

3.1.1 Interpetácia rozsahu cez osu Z

V chápaní reality, ak sa akýkoľvek objekt v reálnom 3D priestore od nás vzdialuje, jeho veľkosť sa opticky zmenšuje, čím je objekt ďalej, tým menší sa nám javí. Naopak, ak sa objekt približuje, jeho veľkosť opticky narastá. To isté platí nielen pre vzdialenosť samotnú, ale aj o vzdialenostiach pohybov v iných smeroch. Napríklad: objekt, ktorý je od nás veľmi vzdialený (javí sa nám ako malý, pretože je ďaleko) a pohybuje sa doľava, vzdialenosť, ktorú prejde v tomto smere sa naším očiam zdá taktiež malá, aj keď v skutočnosti je podstatne väčšia. To isté platí aj pre opačnú situáciu, ak objekt, ktorý je veľmi blízko (zdá sa nám väčší), začne sa pohybovať doľava, táto vzdialenosť sa nám javí ako o mnoho väčšia. Na základe toho, môžeme bezpečne predpokladať, že objekt, ktorý je vzadu sa nachádza v 3D priestore ďalej ako objekt, ktorý je vpredu. Z toho vyplýva rovnako aj, že mierka vzdialeností takýchto objektov je rozdielna, a teda čím je objekt ďalej, tým je táto mierka väčšia, naopak, čím je objekt bližšie, tým je mierka menšia, čo zasa ovplyvňuje aj rýchlosť pohybu objektu pozdĺž osí X a Y. Všetko, čo z toho vyplýva je percentuálne nastavenie rozsahu, na základe ktorého bude prepočítavaný objektový rozsah pre veľkosť, umiestnenie a pohyb. Tieto percentá budú založené na hodnote osi Z – hodnote, ktorá bude vždy určovať ako ďaleko sa nachádza objekt v 3D priestore, ale aj aká je jeho veľkosť pozdĺž tejto osi.

3.1.2 Prekrývanie

Avšak rozsah sám o sebe nemusí byť vždy postačujúci. Ak na scéne existuje viac ako len jeden 3D objekt, pohyb pozdĺž osi Z vzbudzuje pochybnosti. Musíme sa uistiť, že objekty sa navzájom správne prekrývajú, aby sme dokázali správne interpretovať, ktorý objekt je bližšie, tzv. „navrchu“. Ide o vykonanie transpozície alebo prekrývania. Transpozícia, rovnako ako zmena mierky perspektívy je založená na hodnote osi Z. Vo Flashi existuje metóda `swapDepths`, ktorá nám umožní kontrolovať transpozíciu, teda usporiadanie objektov na obrazovke. Táto metóda môže byť použitá so špecifickými objektmi alebo s konkrétnou cieľovou hodnotou. Je to číselná hodnota, ktorá je pre nás v tomto okamihu najdôležitejšia. Vyššia číselná hodnota hĺbky posúva objekt do popredia, teda nad objekty s nižšou hodnotou hĺbky. Vďaka tomu je úplne jednoduché použiť metódu `swapDepths` na usporiadanie objektov pozdĺž osi Z. A práve o tom to všetko je, o instantnej transpozícii. Nanešťastie s týmto súvisí aj jedna komplikácia – vzhľadom k osi Z, ktorú budeme mať, keď pôjdeme ďalej v priestore, teda ďalej od obrazovky. S rastúcou hodnotou osi Z sa bude aj os samotná zväčšovať. Takže osa Z v podstate predstavuje vzdialenosť od diváka. Vyššia hodnota Z predstavuje väčšiu vzdialenosť, a nižšia hodnota predstavuje vzdialenosť menšiu, teda bližšie objekty. Prečo však nemôže byť metóda `swapDepths` použitá priamo s hodnotami osi Z? Vzhľadom na to, že objekty, ktoré sa nachádzajú ďalej majú vyššiu hodnotu hĺbky, mali by teda byť zobrazené navrchu - nad objektmi s hodnotami hĺbky nižšími, ktoré sú však v skutočnosti bližšie. To je vlastne opak toho, čo požadujeme, preto sú metóde `swapDepths` predávané otočené (záporné) hodnoty hĺbky, teda $-Z$. Takto dosiahneme, že pri použití metódy `swapDepths` s hodnotami $-Z$, vzdialenejšie objekty majú hodnoty menšie, a objekty nachádzajúce sa bližšie majú hodnoty väčšie, teda budú naozaj zobrazené pred objektmi s menšími hodnotami.

3.2 Dostupné frameworky pre Flash

Aby nebolo potrebné písať vždy opakujúce sa množstvo procedúr slúžiacich k zobrazovaniu 3D objektov vo Flashi, vzniklo niekoľko frameworkov, ktoré obsahujú knižnice a triedy slúžiace pre uľahčenie práce s nimi. Jedným z najznámejších je `Away3D`, ale nie je to jediný voľne dostupný framework pre vykresľovanie 3D grafiky vo Flashi. Na internete je možné nájsť veľké množstvo podobných frameworkov, ale nie všetky sú bezplatné, a každý z nich má svoje výhody, ale i nevýhody. Ďalej sa budeme zaoberať len tými, ktoré sú k dispozícii zadarmo, v ideálnom prípade sú šírené s otvoreným kódom.

3.2.1 Away3D

`Away3D` je grafický 3D engine [21], ktorý je bezplatný, šírený s otvoreným jadrom. Je písaný pre platformu Adobe Flash v programovacom jazyku `ActionScript 3` a funguje vo všetkých moderných prehliadačoch, ktoré využívajú Adobe Flash Player. `Away3D` umožňuje zobrazovanie 3D modelov z externých zdrojov, priame vytváranie 3D primitív, ako i prevádzanie rôznych operácií a výpočtov v troj-rozmernom priestore. Jeho jadro bolo založené na staršom a menej populárnom grafickom engine `Papervision3D`, ktorý bol postavený práve na platforme Adobe Flash. `Away3D` podporuje hierarchické transformácie s objektmi prostredníctvom funkcií ako sú: pozícia, rotácia, veľkostná mierka, zobrazovanie bitmapových aj video textúr alebo osvetlenie v reálnom čase s využitím viacerých spôsobov tieňovania, či niekoľko druhov kamier umožňujúcich rôzne pohľady a pokročilú prácu so scénou. `Away3D` je súčasne vyvíjaný vo viacerých vetvách a to `Away3D` pre Flash Player 10, ktorý ponúka aj obmedzené možnosti pre využitie akcelerácie za pomoci grafickej karty (GPU),

ďalej pre Flash Player 9, a nakoniec Away3D Lite, kompaktnú verziu, ktorá neponúka všetky možnosti, ale menšiu veľkosť a zrýchlenie za pomoci optimalizácie na úkor obmedzených možností 3D zobrazovania, všetky verzie sú platformovo nezávislé, takže týmto smerom voľba nie je taktiež obmedzená.

3.2.2 PaperVision3D

Ďalším dostupným 3D engine-om je PaperVision3D (ďalej len PV3D) rovnako šírený s otvoreným jadrom vytvorený pre platformu Flash. Oficiálne podporuje len Flash Player do verzie 9, v ktorom je využívaný jazyk ActionScript 3, pri použití jazyka ActionScript 2 je postačujúci Flash Player vo verzii 8. Odporúčané je využitie jazyka AS 3, pretože s ním PV3D pracuje rýchlejšie a je stabilnejší. Aj keď tím vývojárov PV3D ohlásil práce na novej verzii tohto engine-u, ani do tejto chvíle sa im nepodarilo zaviesť plnú podporu pre Flash Player 10. Medzi jeho výhody patrí: lineárne mapovanie textúr, tieňovanie, podpora rôznych materiálov, optimalizácia pre kvalitu a rýchlosť vykresľovania. PV3D je navrhnutý tak, aby jeho používanie bolo ľahké a jednoduché.

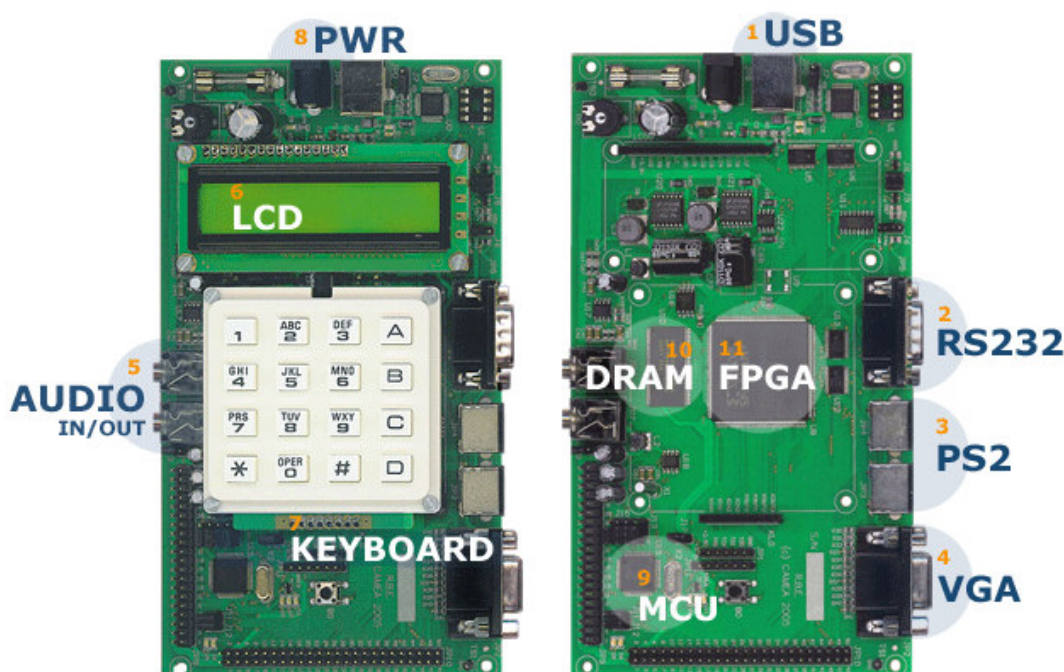
3.2.3 Flash Sandy 3D engine

Flash Sandy je objektovo-orientovaný engine umožňujúci prácu v 3D prostredí platformy Flash, ktorý je opäť šírený s otvoreným jadrom. Dokáže využívať ako jazyk ActionScript 2, tak ak 3, v závislosti používanej verzie jazyka je vyžadovaná aj podpora vyššej verzie Flash Playera. Medzi hlavné rysy Flash Sandy patrí: informácie o objeme orezávania pre perfektné scény interiéru, pokročilé efekty tieňov, systém materiálov umožňujúci jednoduchú zmenu povrchu objektov, vrátane materiálov umožňujúcich vytvoriť priehľadné plochy, bitmapové a video textúry, pokročilá správa kamery, kompatibilita s Flash Playerom od verzie 7 až do verzie 10, kompletná sada parserov pre obsluhu rôznych 3D formátov, či niekoľko 3D primitív, ktoré umožňujú rýchle a jednoduché vytvorenie 3D objektov s voliteľnými parametrami bez znalostí 3D modelovania [22].

4 Návrh aplikácie

4.1 Špecifikácia

Cieľom práce je vytvorenie interaktívnej aplikácie, ktorá dovolí prezentovanie FITkitu a jeho súčastí na internetových stránkach. Po preskúmaní dostupných technológií, a bližšom sa zoznámení s nami zvolenou technológiou, ktorú vyberieme ako najvhodnejšiu, bude s využitím vhodného modelovacieho nástroja potrebné vytvoriť 3D model FITkitu, ktorý bude zložený z niekoľkých submodelov. Navrhovaný je bezplatný editor Blender. Aplikácia by mala umožňovať náhľad na platformu minimálne z dvoch pohľadov, a to pohľad zhora a pohľad na plošný spoj zo strany súčiastok. Je potrebné zvoliť aj vhodnú formu zvýraznenia jednotlivých komponentov platformy tak, aby používateľ dokázal už na prvý pohľad rozlíšiť podstatné časti FITkitu. Odporúčaným riešením bolo prevzatie frameworku Flash Sandy s využitím univerzálneho programovacieho jazyka HaXe. Z dôvodu možnej rozšíriteľnosti aplikácie v budúcnosti je požadované využitie bezplatného softvéru, v ideálnom prípade takého, ktorý je šírený s otvoreným jadrom.



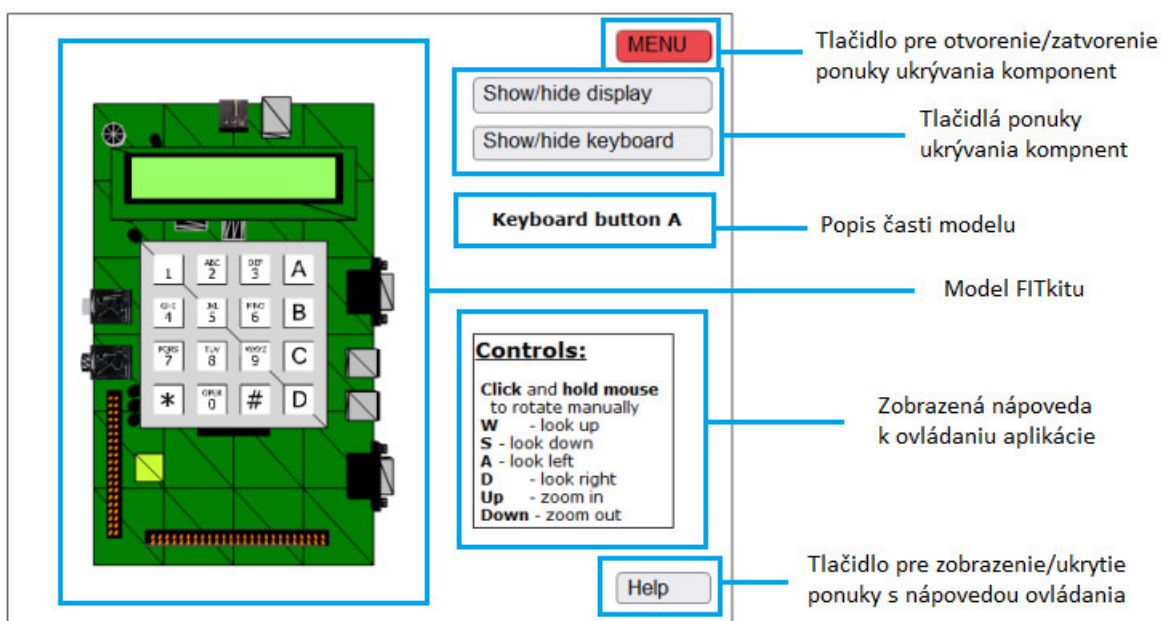
Obrázok 4.1 – Platforma FITkit (pohľad s klávesnicou (K) a displejom (D), pohľad bez K a D) [1]

4.2 Navrhované riešenie

Interaktívna aplikácia slúžiaca na prezentáciu FITkitu bude založená na platforme Flash. Následne bude potrebné zvoliť jeden z dostupných frameworkov, v ktorom budeme aplikáciu vyvíjať. Tvorbu aplikácie rozdelíme do jednotlivých krokov, pričom prvým krokom bude vytvorenie 3D modelu. Každú požadovanú komponentu FITkitu bude predstavovať samostatný submodel, pričom výsledný model bude zložením týchto submodelov do jedného celku. Následne bude celý model exportovaný

do formátu, ktorý bude zvolený framework schopný importovať, a správne zobrazíť model ako celok, ale zároveň bude schopný pracovať aj samostatne s jednotlivými submodelmi. Keďže pre modelovací nástroj Blender je dostupný zásuvný modul, ktorý umožňuje export objektov ako objekty programovacieho jazyka ActionScript 3.0, s ktorými dokážu pracovať všetky dostupné frameworky, už v tomto kroku nahradíme navrhovaný jazyk HaXe jazykom ActionScript. Tým pádom už nebude potrebné riešiť importovanie modelu do frameworku, pretože frameworky dokážu pracovať priamo s jazykom ActionScript, a teda aj s objektmi (modelmi) uloženými v tomto formáte.

Druhým krokom bude naprogramovanie samotnej aplikácie zobrazujúcej model, pričom rozloženie na jednotlivé submodely nám umožní tieto objekty zo scény odstrániť a späťne ich pridávať, čím dosiahneme efekt „rozoberania“ FITkitu. Koncový používateľ tak bude mať možnosť vidieť aj komponenty, ktoré nie sú na prvý pohľad viditeľné, pretože sa nachádzajú ukryté pod nejakým iným komponentom. Model si bude môcť používateľ ľubovoľne približovať, oddiaľovať, vo všetkých smeroch otáčať, takže dosiahneme v podstate neobmedzené množstvo pohľadov, čím vytvoríme dojem ako keby používateľ FITkit reálne držal v rukách a prezeral si ho. Ruky však bude potrebovať len na ovládanie aplikácie, ktorú bude možné ovládať ako myšou, tak aj klávesnicou. Priblíženie sa k realite - dôveryhodnosť celej aplikácie bude dotvorená použitím reálnych textúr. Je potrebné zvoliť aj vhodný spôsob zvýraznenia jednotlivých komponent, aby používateľ dokázal rozlíšiť rôzne súčasti modelu.



Obrázok 4.2 – Návrh rozhrania aplikácie

Ako vidíme na obrázku 4.2, aplikácie je navrhnutá tak, aby používateľom ponúkala intuitívne rozhranie. V ľavej časti bude umiestnený samotný model FITkitu, v pravom hornom rohu bude tlačidlo „MENU“, ktoré bude slúžiť k zobrazeniu a ukrytiu ponuky pre skrývanie jednotlivých komponent, pričom táto ponuka sa bude skladať z ďalších tlačidiel, z ktorých každé bude plniť funkciu odstránenia, resp. pridania danej komponenty zo/do scény. Ponuka sa bude zobrazovať pod tlačidlom „MENU“. V pravom dolnom rohu bude umiestnené tlačidlo „Help“, ktoré bude slúžiť k zobrazeniu alebo ukrytiu ponuky s návodou k ovládaniu aplikácie. Napravo od modelu medzi jednotlivými ponukami sa bude zobrazovať podrobnejší popis komponenty, nad ktorou sa bude práve nachádzať kurzor myši.

5 Voľba použitej technológie

Nasledujúca kapitola a jej podkapitoly čerpajú z [23] a [24].

Budeme porovnávať 3 najznámejšie a najviac rozšírené frameworky, medzi ktoré patria Away3D, Flash Sandy a PaperVision3D (PV3D). Na fóre spoločnosti Adobe [25] rôzni používatelia zdieľujú svoje názory a skúsenosti, a porovnávajú práve tieto 3 frameworky. Všetky 3 frameworky dokážu vytvárať a zobrazovať 3D objekty v reálnom čase priamo v okne internetového prehliadača pomocou technológie Flash. Návštevníkom internetových stránok tak prinášajú pôsobivý vizuálny zážitok a používateľovi umožňujú priamu interakciu.

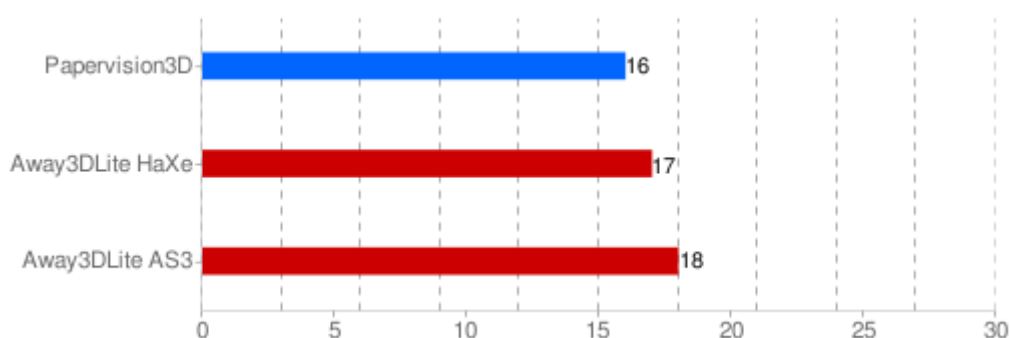
Prvé dva sú dostupné hneď v troch verziách a to s podporou jazyka ActionScript 2, ActionScript 3 a HaXe. PV3D podporuje len jazyk ActionScript. Najstarším frameworkom je Flash Sandy, o niečo mladším je PV3D, a najmladším je Away3D. Zaujímavosťou je, že Away3D začal vznikať ako rozšírenie pôvodnej, už staršej verzie PV3D 1.5, pri ktorej sa vývojári dožadovali viacerých chýbajúcich funkcií, ale priniesol také množstvo rozšírení, až sa nakoniec úplne oddelil, a pokračoval vo svojom vývoji nezávisle.

5.1 Porovnanie PaperVision3D a Away3D

5.1.1 Popis testov

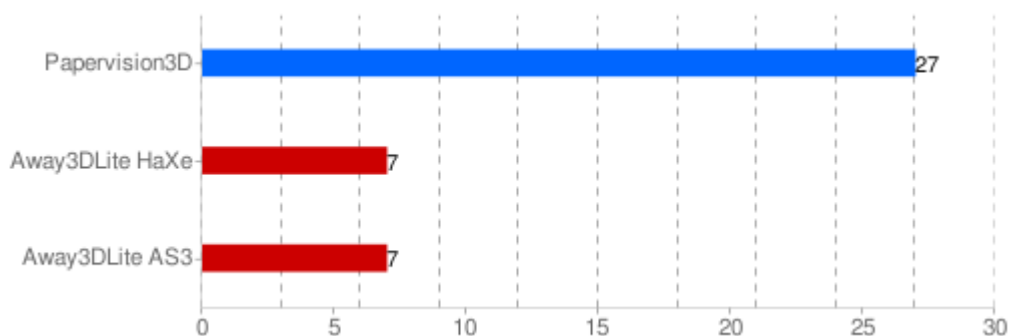
Nasledujúce testy porovnávajú výkonnosť a využitie prostriedkov frameworku PV3D a Away3D, ktorý bol testovaný v dvoch verziách, a to ako s využitím jazyka HaXe, tak aj AS 3.0. Testy boli vykonané pod operačným systémom Windows XP SP3, s využitím prehrávača Flash Player vo verzii 10 v internetovom prehliadači Mozilla Firefox 3.6.3. Scéna obsahuje modely s vysokými detailmi, počet polygónov v každom snímku je okolo 1340, výsledná scéna obsahuje celkovo niečo málo cez 3000 polygónov. Vzhľadom k tomu, že možnosti oboch frameworkov sú odlišné, boli použité len jednoduché farebné materiály objektov, ktoré obsahujú oba frameworky. Testy boli prevzaté z [23].

5.1.2 Výsledky



Obrázok 5.1 – Porovnanie výkonnosti, počet snímkov za sekundu (fps) [23]

Na obrázku 5.1 môžeme vidieť porovnanie pri vyrenderovaní totožnej scény, z ktorého vyplýva, že Away3D si v porovnaní s PV3D naozaj polepšilo, a vďaka lepšej optimalizácii získava mierny výkonnostný náskok. Najlepšie výsledky dosahuje verzia využívajúca ActionScript3.



Obrázok 5.2 – Porovnanie výkonnosti, využitie pamäte (v MB) [23]

Na obrázku 5.2 sa rozdiel pri porovnaní využitia operačnej pamäte ale podstatne zväčšuje. Zatiaľ čo PV3D si ukrojil až 27MB, Away3D si v oboch prípadoch vyžiadal len 7MB pamäte. Toto je už nezanedbateľný rozdiel, ktorý by sa s počtom pridaných objektov do scény len zväčšoval. Away3D jasne ukazuje, kde sú slabé miesta jeho predchodcu, a že vývojári si pri jeho ladení dali naozaj záležať, nové funkcie implementujú takmer nepretržite.

Away3D sa tak stáva hybnou silou, keď poukazuje na množstvo nedostatkov, ktoré sa vývojárom PV3D stále nepodarilo opraviť napriek tomu, že niektoré nové funkcie už prebrali z Away3D frameworku. Pri vývoji Away3D bol kladený veľký dôraz na výkonnosť a jednoduchosť použitia, o čom svedčí aj rastúca komunita a ohlasy používateľov. PV3D stráca v porovnaní s Away3D po všetkých stránkach, vrátane kvality zobrazovania a hĺbkového triedenia. Keďže chceme, aby naša aplikácia ponúkala používateľom čo najväčší komfort, a zároveň boli jej nároky na systémové prostriedky minimalizované, musíme PV3D ako jednu z možností vylúčiť.

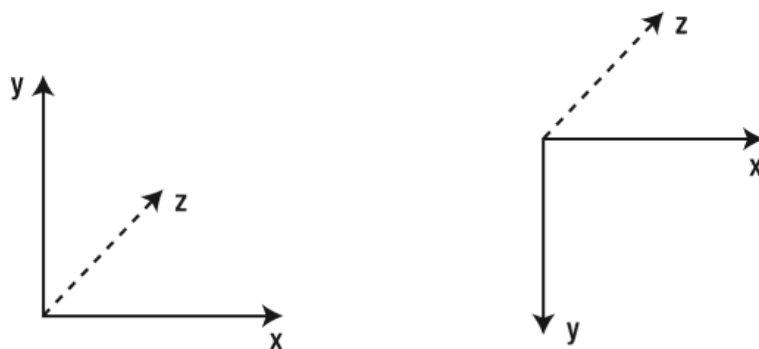
5.2 Porovnanie Away3D a Flash Sandy

Framework Flash Sandy nie je síce tak vychvaľovaný a populárny ako jeho konkurencia, ale jeho dôkladná funkcionálna a schopnosť zobrazovania 3D scén je s nimi porovnateľná. Veľkou výhodou je veľmi dobrá dokumentácia jeho knižníc a tried, s pomerne dostatočným množstvom príkladov. Flash Sandy taktiež disponuje niektorými funkciami, ktoré chýbajú jeho konkurencii. Medzi veľké negatíva ale patrí veľmi výrazné spomalenie v dôsledku pridávania viacerých objektov do scény, ktoré je oveľa výraznejšie ako v prípade Away3D, čo je spôsobené použitím vektorového vykresľovania pôvodného Flashu, ktoré Flash Sandy využíva. Boli urobené aj testy so skutočným rastrovaním, ale výsledky boli príliš pomalé pre cieľové rozlíšenie.

Flash Sandy ako prvý implementoval aj interakciu pomocou myši s 3D objektmi, projekciu textúr, ukladanie povrchových materiálov do medzipamäte, či perspektívne správnych materiálov. Významným prínosom bolo aj zavedenie LOD [26] (Level of detail) - úrovne detailov objektov, ktorá umožňuje zníženie zložitosti reprezentácie 3D objektu v závislosti na vzdialenosti od koncového diváka alebo iných ukazovateľov, ako je napr. dôležitosť zobrazovaného objektu, rýchlosť pohybu alebo poloha. Táto technika zvyšuje efektívnosť zobrazovania znížením záťaže grafickej scény, zvyčajne pomocou vertexových transformácií.

5.2.1 Problém z-sortingu

Flash Sandy disponuje len základným triedením hĺbky objektov, ktoré ponúka priemerné výsledky, zatiaľ čo Away3D ponúka 3 spôsoby vykresľovania. Z-sorting (alebo tiež z-ordering) je usporiadanie pri prekrývaní dvoch alebo viacerých, v našom prípade troj-rozmerných objektov v aplikácii, keď jeden objekt môže prekrývať celý alebo len časť iného objektu. Tento termín sa vzťahuje na usporiadanie objektov pozdĺž osi Z. V súradnicovej geometrii X typicky odkazuje na vodorovnú os, Y na vertikálnu os a Z označuje os, ktorá je kolmá na osi X a Y (obrázok 5.3). Pre správne zobrazenie prekrývania 3D objektov je potrebné tieto objekty správne zotriediť, čím sa môžu niektoré objekty v scéne javiť ako neviditeľné, pretože sú celé prekryté iným objektom. Táto neviditeľnosť objektu poskytuje výhodu pri vykresľovaní scény, pretože taký objekt nemusí byť vôbec na scéne zobrazený, hardvér počítača tak vykreslenie celej scény vypočíta rýchlejšie. [27] Z-sorting bol až donedávna jedným z najväčších problémov 3D frameworkov, ktorý sa ako prvému podarilo vyriešiť a ponúknuť správnu implementáciu práve tímu vývojárov Away3d.



Obrázok 5.3 – Systém súradníc v Away3D (vľavo) v porovnaní s natívnym zobrazovaním súradnicového systému vo Flashi (vpravo)

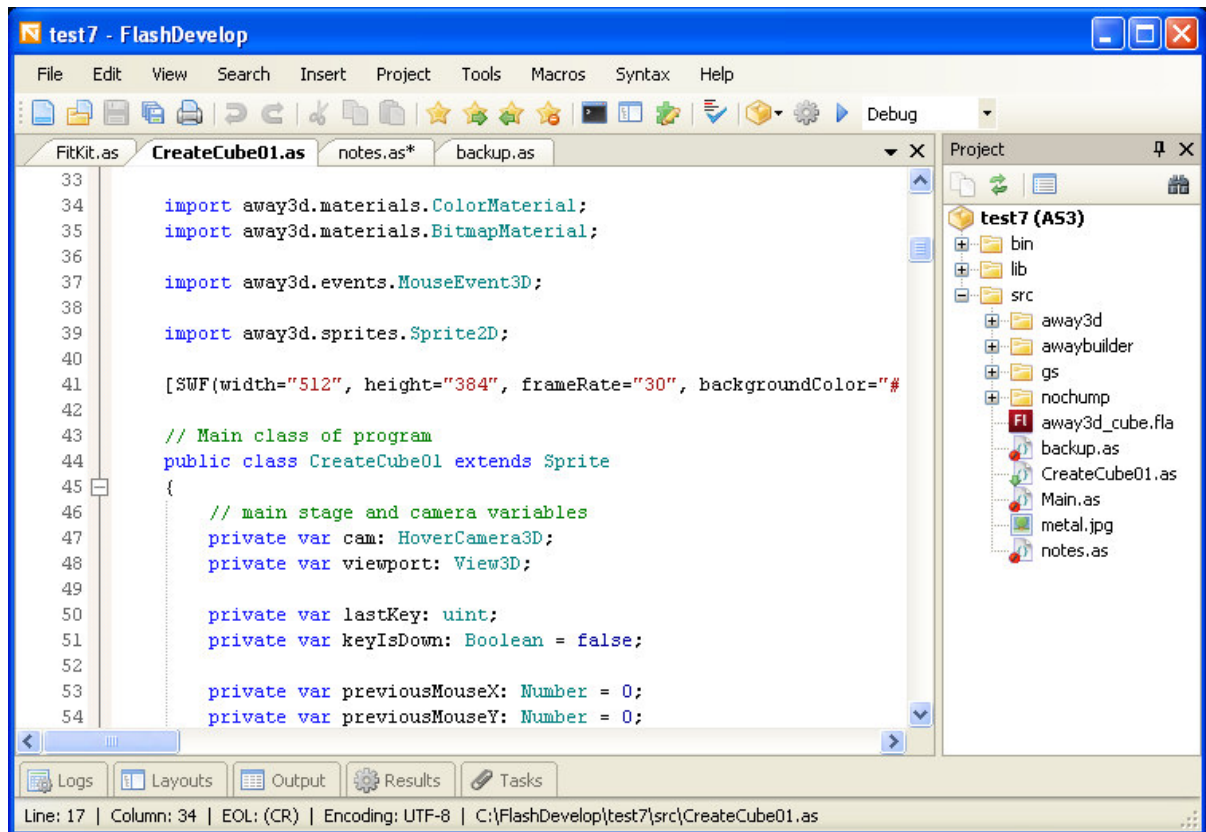
5.3 Výsledky

Away3D tak z tohto súboja vychádza ako najmodernejší framework súčasnosti, ktorý aj napriek svojim nedostatkom prekonáva konkurenciu. V našom teste je najmocnejším nástrojom ponúkajúcim dostupnosť 3D rozhrania zadarmo a pre všetkých, čo nám umožňuje aj v budúcnosti rozširovať funkcionality aplikácie a tvorbu dem. Okrem nadštandardných funkcií je jeho veľkou výhodou aj kvalitná dokumentácia s ukázkovými príkladmi. Vzhľadom na jednoduchosť práce s ním, ktorej dopomáha aj množstvo dostupných on-line návodov, sa stáva jasnou voľbou pri vývoji interaktívnej 3D aplikácie v prostredí internetu.

5.4 Bližší pohľad na framework Away3D

Zo spomínaných verzií frameworku Away3D (kapitola 4.2.1), naše požiadavky najlepšie spĺňa najvyššia verzia pre FP10, ktorá nám umožňuje využiť všetky prednosti engine-u, a zároveň aj využitie najnovších technológií 3D zobrazovania. Pri vývoji aplikácie bola použitá verzia 3.6.0, ktorú je možné stiahnuť z oficiálnych stránok engine-u [28], kde je k dispozícii na stiahnutie aj množstvo príkladov pre nami zvolený editor FlashDevelop (obrázok 5.4), ako aj pre iné editory vrátane Adobe Flash CS Professional, Flash Builder, či Adobe Flex.

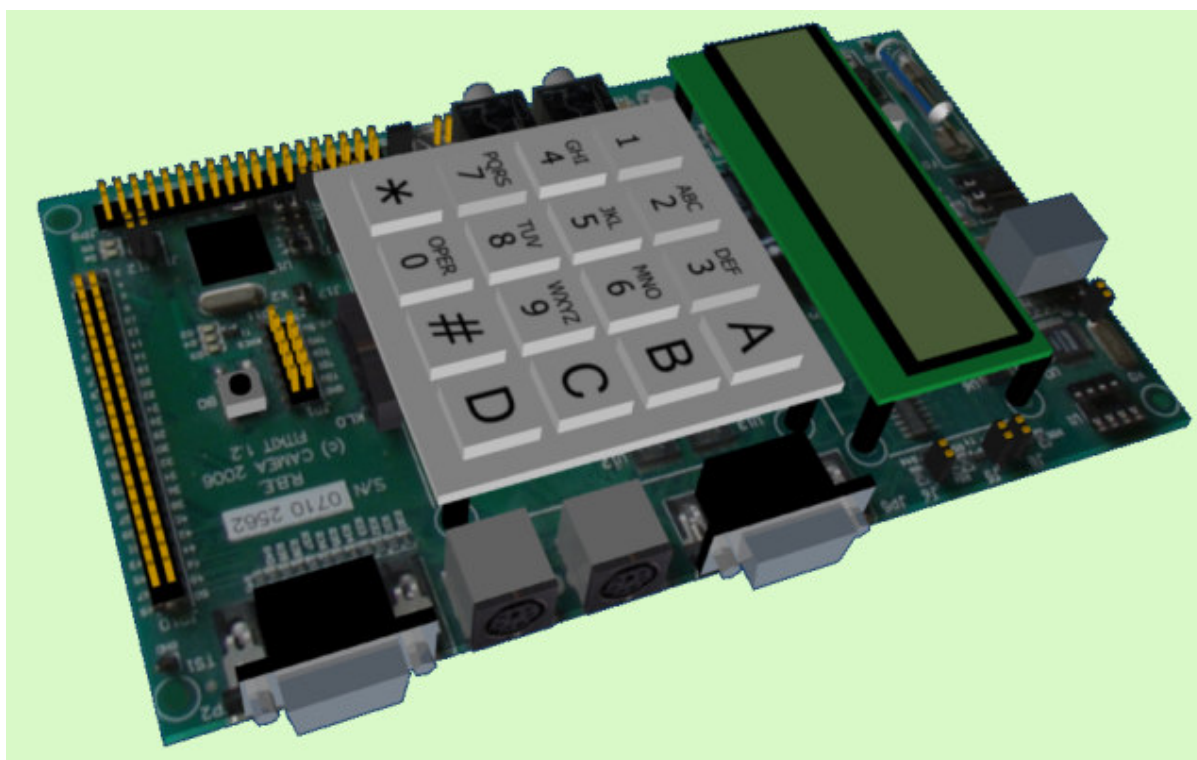
Návod na správne nastavenie nami zvoleného editora FlashDevelop pre spoluprácu s knižnicami grafického engine-u Away3D nájdeme na internetových stránkach spolu aj s inými tutoriálmi [29].



Obrázok 5.4 – Zvýraznenie syntaxe v editore FlashDevelop s importovanou knižnicou Away3D

6 Tvorba 3D modelu

Model bude vytvorený v modelovacom nástroji Blender. Základným objektom celého modelu je matičná doska, ku ktorej sú pripojené všetky ostatné prvky FITkitu. Ďalšími dôležitými súčastami sú LCD displej a klávesnica, ktorá sa skladá zo 16 tlačidiel. Tieto dve komponenty sú k matičnej doske pripevnené pomocou 4 držiakov, v každom rohu jeden. Klávesnica ako aj displej sú s matičnou doskou prepojené pomocou niekoľko-pinových konektorov. Naľavo od klávesnice sa nachádzajú dva rovnaké audio porty, jeden pre vstup a druhý pre výstup. Napravo od klávesnice sa nachádzajú dva porty PS2, jeden slúži na pripojenie klávesnice, a druhý na pripojenie myši. Nad týmito portami sa nachádza konektor rozhrania RS232, ktorý je totožný s VGA konektorom, ktorý sa zasa nachádza pod portami PS2. V strede modelu sa nad displejom nachádza konektor externého napájania a USB port slúžiaci na pripojenie FITkitu. Pod klávesnicou sa ukrýva operačná pamäť typu DRAM a FPGA čip typu Spartan 3. Na ľavej strane o niečo nižšie ako klávesnica sa nachádza procesor MCU rodiny Texas Instruments. V strede dolnej polovice sa nachádza jumper. Na matičnej doske sa nachádza aj niekoľko odporov (kondenzátorov) a konektorov DIN. Reálny vzhľad modelu bude dotvorený použitím textúr matičnej dosky, čipu FPGA, pamäte DRAM, jednotlivých portov a konektorov, ako aj tlačidiel klávesnice a odporov. Rozmiestnenie komponentov zobrazuje obrázok 6.1.



Obrázok 6.1 – Model FITkitu vytvorený v modelovacom nástroji Blender

6.1 Vloženie 3D modelu do scény

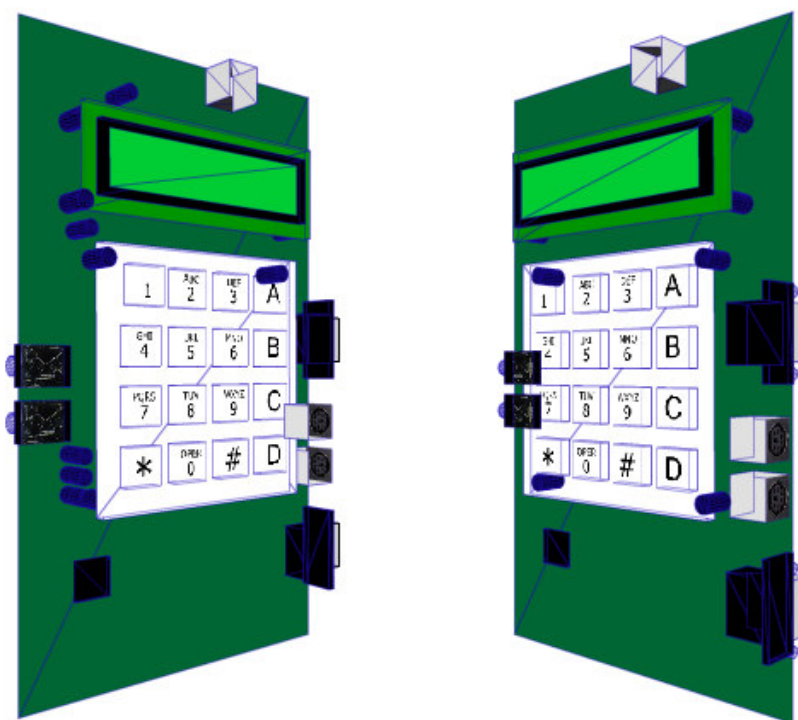
Po tom, čo sme s využitím zásuvného modulu výsledný model z programu Blender vyexportovali do formátu súborov AS3, sme tento model následne skúsili importovať do scény, ale ako pôvodný

framework Flash Sandy, tak aj Away3D mali s týmto formátom drobné problémy, keď niektoré polygóny neboli vykreslené správne, a v objektoch vznikali diery alebo iné nežiadúce „efekty“, bol tento spôsob vloženia modelu do scény neprípustný, pretože výsledný FITkit na scéne vyzeral inak ako v prostredí programu Blender, a teda aj inak ako bolo požadované.

Bolo teda nevyhnutné hľadať iné riešenie, ktorým sa stalo vytvorenie FITkitu s využitím dostupných primitív, ktoré ponúka cieľový framework.

6.2 Overenie výsledkov testu v praxi

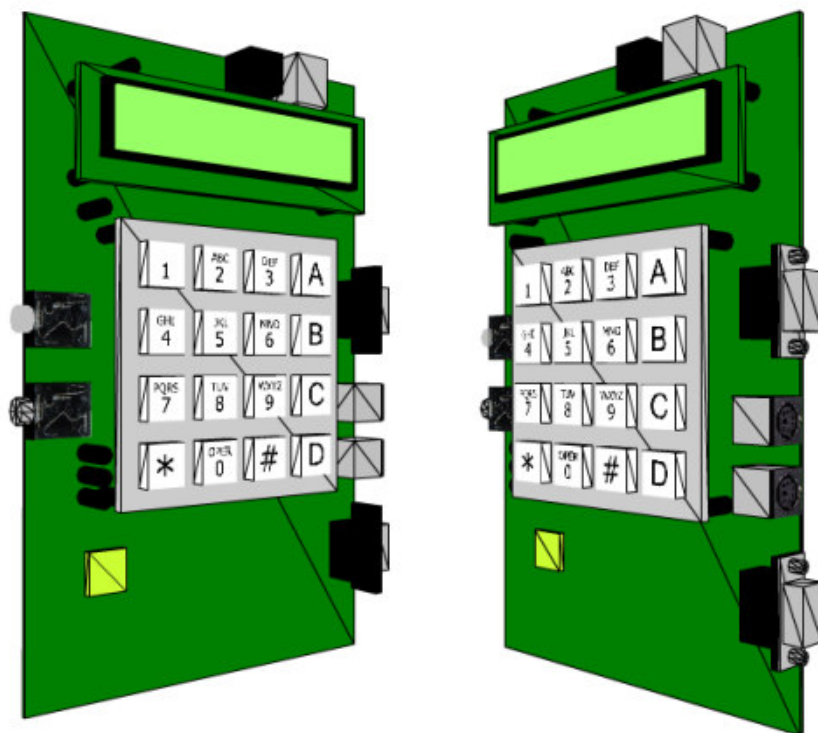
Aby sme sa presvedčili o možnostiach dvoch (PV3D môžeme vynechať, pretože Away3D vychádza práve z neho) súperiacich frameworkov, ktorými sú Away3D a Flash Sandy aj v praxi, vytvoríme jednoduchú scénu, ktorá bude zobrazovať počítačový koncept požadovaného modelu. Tento model vytvoríme len s využitím dostupných primitív. Cieľom bude vytvoriť, čo najviac podobné scény, ktorých rozdiely by sa už mali prejavovať v praxi.



Obrázok 6.2 – Koncept modelu vytvorený vo frameworku Flash Sandy (pohľady zľava a sprava)

Ako vidíme na obrázkoch 6.2 a 6.3, modely sú naozaj takmer totožné, čo sa týka počtu a rozloženia objektov. Obidva modely obsahujú textúry jednotlivých tlačidiel klávesnice, ani v jednom z nich v tomto prípade nenastal žiadny problém. Zďaleka podobné však nie sú výsledky zobrazených scén, pretože framework Away3D využíva jadro so správnym z-sortingom, zatiaľ čo Flash Sandy takouto možnosťou nedisponuje a prekrývanie objektov na scéne je nesprávne, priam katastrofálne nevhodné. Nesprávne prekrývanie objektov je viditeľné na ľavej strane obrázkov, kde si môžeme všimnúť, že v prípade Flash Sandy sú porty pre PS2 zariadenia vykreslené veľmi záhadne, keď jeden port klávesnicu prekrýva, zatiaľ čo druhý nie. Podobný efekt môžeme vidieť u držiakov podstáv klávesnice a displeja. Ďalším nevysvetliteľným faktom je rozdielne prekrývanie pri pohľade

na model z pravej strany, keď vidíme, že už len samotný Flash Sandy zobrazuje prekryvanie objektov úplne odlišne ako v predchádzajúcom prípade, keď boli nesprávne prekreslené len niektoré držiaky podstav ako u klávesnice, tak aj displeja. V tomto prípade sú držiaky displeja prekryvané správne, ale naopak pribudli nesprávne sa prekryvajúce držiaky klávesnice. Nesprávne sa prekryvajú aj audio porty, a pri vykresľovaní PS2 portov dochádza k zaujímavému úkazu, keď vidíme aj polygóny vo vnútri objektu. Away3D tak potvrdzuje, že je naozaj tou najlepšou voľbou.



Obrázok 6.3 –Koncept modelu vytvorený vo frameworku Away3D (pohľady zľava a sprava)

6.3 Sprevádzkovanie Away3D a AS 3.0 v prostredí FlashDevelop

V prostredí programu FlashDevelop si nastavíme cesty ku knižniciam, ktoré je potrebné nastaviť v menu Tools - Program settings, prepne sa na záložku AS3Context, pretože chceme využívať možnosti jazyka AS 3.0, tu je potrebné v riadku „Flex SDK Location” nastaviť správnu cestu ku priečinku, v ktorom máme nahrať framework Flex SDK.

6.4 Rozdelenie aplikácie

Každý projekt vytvorený v prostredí FlashDevelop je rozdelený do niekoľkých priečinkov. V priečinku „bin“ sa nachádzajú spustiteľné súbory, ktoré vzniknú po kompilácii, do priečinka „lib“ môžeme nahrávať ľubovoľné súbory, ktoré chceme v našom projekte používať, napríklad fonty, textúry, modely a pod. Najdôležitejší je pre nás priečinok „src“, ktorý obsahuje všetky súbory so zdrojovými kódmi, vrátane podpriečinka „away3d“, v ktorom sú uložené všetky súbory grafického engine-u, a po otvorení projektu sú automaticky načítané. Všetky prostriedky, ktoré engine ponúka sú

pri vytváraní aplikácie k dispozícii, rovnako aj pri kompilácii je obsah knižníc automaticky vložený do výsledného spustiteľného súboru, čo aplikáciu robí platformovo nezávislou.

6.5 Prvá aplikácia s využitím Away3D

Všetko si názorne vysvetlíme na kóde z ukážky 6.1 (príklad prevzatý z [30]):

```
1 package {
2     import away3d.containers.View3D;
3     import away3d.primitives.Sphere;
4     import flash.display.Sprite;
5     public class Basic01 extends Sprite {
6         public function Basic01() {
7             var view:View3D = new View3D({x:200,y:200});
8             addChild(view);
9             var sphere:Sphere = new Sphere();
10            view.scene.addChild(sphere);
11            view.render();
12        }
13    }
14 }
```

Ukážka 6.1 – Prvý program s využitím frameworku Away3D

Riadok č. 1: AS vyžaduje, aby všetky triedy boli v balíčku (package), v tomto príklade používame nepomenovaný balíček (inak označovaný tiež ako predvolený).

Riadok č. 2-4: Aby sme mohli používať triedy frameworku Away3D, musíme ich importovať, rovnako aj triedy knižníc využívajúcich funkcionality technológie Flash.

Riadok č. 5: Našu triedu sme pomenovali Basic01, táto trieda je rozšírením triedy Sprite, čo znamená, že naša trieda môže vykonávať všetky funkcie ako trieda Sprite.

Riadok č. 6: Funkcia, ktorá má rovnaký názov ako trieda sa nazýva „konštrukčná“ funkcia. Vždy, keď bude vytvorená nová inštancia tejto triedy, vykoná sa obsah tejto funkcie (Basic01).

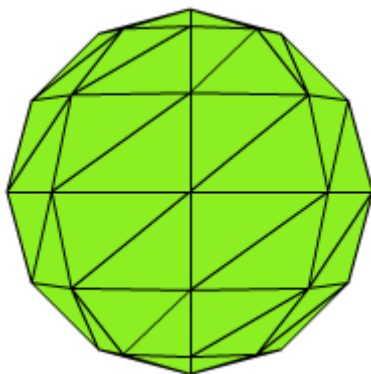
Riadok č. 7-8: Vytvoríme nové zobrazenie (3D scénu s názvom view), ktorá bude našim pohľadom do trojrozmerného sveta. Parametre „x“ a „y“ povedia engine-u, kde má nastaviť stred nášho sveta. Následne pridaním tohto pohľadu do našej triedy, zabezpečíme, že bude pridaný do flashového obsahu.

Riadok č. 9-10: Vytvoríme niečo, čo by sme mohli v našom svete vidieť - 3D objekt gule (sphere). Potom sme pridalí túto guľu do predvolenej scény nášho zobrazenia, aby bola viditeľná. Keďže sme nenastavili žiadne parametre x, y, z pre guľu, bude vytvorená s počiatočným nastavením týchto parametrov (x = 0, y = 0, z = 0).

Riadok č. 11: Away3D nám umožňuje rozhodnúť sa, kedy chceme výslednú scénu vykresliť (renderovať). Renderovanie je proces prevodu virtuálneho 3D obsahu do viditeľného 2D obrazu.

Renderovanie je náročné na výpočetný čas procesora, preto je veľmi užitočné, že môžeme ovplyvniť, kedy chceme výslednú scénu aktualizovať, čiže vykresliť znova.

Ako bude vyzerat' výsledok tohto jednoduchého programu z ukážky 6.1 po spustení, môžeme vidieť na obrázku 6.4.



Obrázok 6.4 – Výsledná scéna po vyrenderovaní a spustení aplikácie

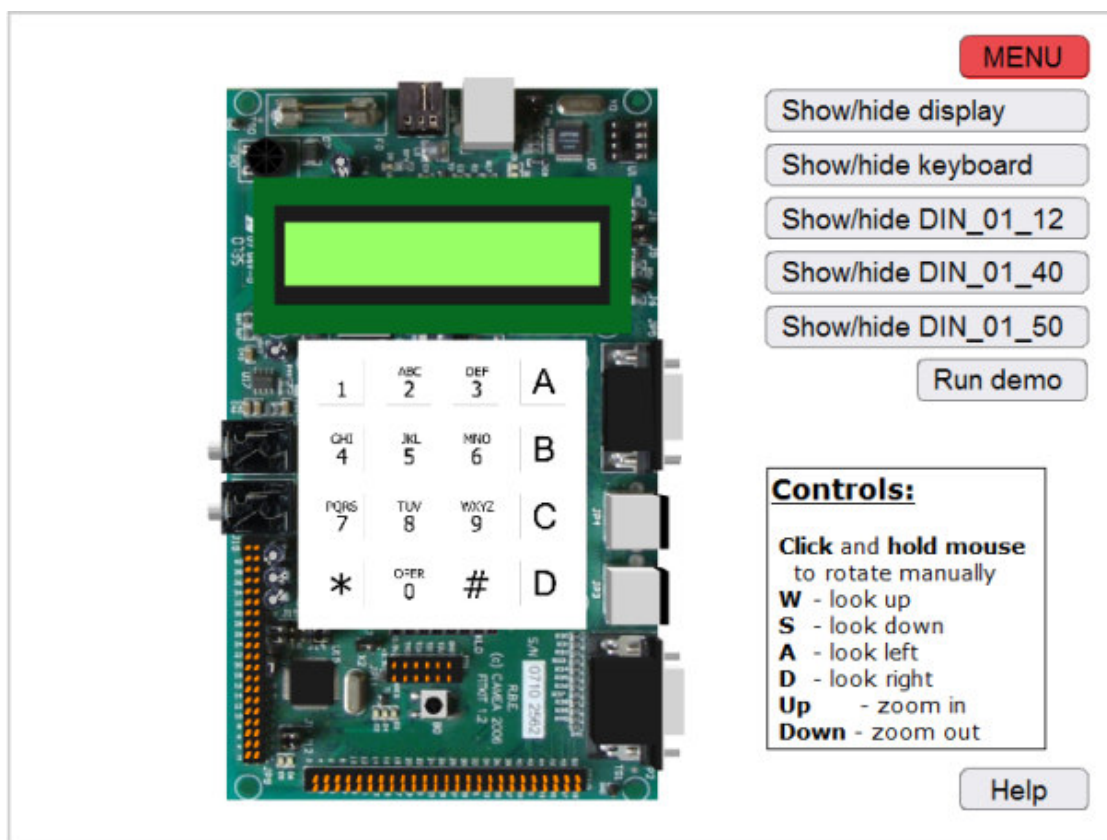
Výsledkom po kompilácii a vyrenderovaní je scéna, v ktorej strede sa nachádza 3D objekt – guľa.

7 Implementácia

Pri vytváraní aplikácie bol použitý už spomínaný jazyk ActionScript 3.0 v spolupráci s frameworkom Away3D, pričom celá aplikácia bola vytváraná v prostredí programu FlashDevelop. Aby bolo možné využívať aj vlastnosti vektorovej grafiky platformy Flash, aplikácia bola kompilovaná s knižnicami Flex SDK.

7.1 Model

Ako sme už spomínali v kapitole 6.2, pri vytváraní modelu sme odstúpili od modelu vytváraného v modelovacom nástroji Blender, a prešli sme k tvorbe pomocou primitív. Jednotlivé komponenty FITkitu sú vytvorené z primitívnych telies akými sú kocka, kváder či valec. Týmto objektom sú pomocou parametrov nastavené špecifické rozmery, aby verne odrážali pomer voči reálnej platforme. Ako môžeme vidieť, výsledný model (obrázok 7.1) vytvorený pomocou primitív je v mnohých detailoch takmer totožný s modelom vytvoreným v Blenderi (obrázok 6.1), čím sme dokázali, že ani pri vytváraní modelu priamo v prostredí frameworku sme sa nemuseli obmedzovať, či výsledný model nejakým spôsobom „ochudobniť“. Povrch týchto objektov tvorí prevažne jedna farba (ColorMaterial), veľké množstvo objektov má na niektorej zo svojich strán nanesenú textúru, ale na niektoré objekty je nanesený špeciálny materiál (ShadedMaterial), ktorý umožňuje vytvoriť efekt tieňovania pri rôznych uhloch pohľadu na objekt, pričom tento efekt je ovplyvnený dopadom svetla z bodového zdroja (DirectionalLight3D), ktoré svieti len v zadanom smere.



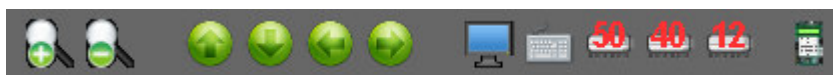
Obrázok 7.1 – Výsledná podoba aplikácie, zobrazená ponuka menu a nápovedy

Základom výsledného modelu je matičná doska, na ktorú sú „pripojené“ ostatné komponenty, medzi ktoré radíme napájací konektor, port pre pripojenie prostredníctvom rozhrania USB, porty pre vstup a výstup zvukového rozhrania, porty typu PS2 umožňujúce pripojenie myši a klávesnice, konektor VGA umožňujúci grafický výstup na obrazovku externého monitora, konektor RS232, riadkový LCD displej, ktorý je pripojený pomocou 4 držiakov, klávesnica pozostávajúca zo 16 kláves, ktorá je rovnako pripojená pomocou 4 držiakov, pamäť typu DRAM, čip FPGA, procesor MCU rodiny MSP430, konektory pre pripojenie klávesnice a displeja k matičnej doske, 3 DIN konektory rôznych veľkostí, čipy U20 i U21, 5 menších a 2 väčšie odpory (kondenzátory), trimr P0 a nakoniec jumpery.

7.2 Interaktivita

Ovládanie aplikácie je vskutku intuitívne, pretože manipulovať s modelom môže používateľ prostredníctvom myši, a to tak, že klikne kdekoľvek do okna aplikácie, a pri stlačení tlačidla pohybuje myšou, čím zároveň rotuje model v požadovanom smere. Manipulovať modelom je používateľovi taktiež umožnené aj pomocou klávesnice, stlačením kláves W, A, S, D. Ďalej je možné si model približovať/oddiaľovať stlačením kláves šípka hore/šípka dole. Tlačidlo MENU v pravom hornom rohu otvára kontextovú ponuku, ktorej jednotlivé tlačidlá ukrývajú/zobrazujú špecifické komponenty ako súčasť modelu. Posledné tlačidlo umožňuje spustenie krátkeho dema, ktoré slúži na rýchlu prezentáciu platformy FITkit, a zároveň demonštruje niektoré možnosti manipulácie s modelom. Vpravo vedľa modelu je zobrazovaný popis komponenty, nad ktorou sa práve nachádza ukazovateľ myši, v prípade, že sa kurzor dostane mimo model, je tento popis automaticky ukrytý. Posledné tlačidlo vpravo dole zasa zobrazí nápovedu k ovládaniu aplikácie. Pre lepšiu prehľadnosť je nápoveda automaticky ukrytá, ak sa kurzor objaví nad modelom, pretože v prípade komponent, ktoré majú viac-riadkový popis by sa druhý a ďalšie riadky prekryvali s touto ponukou. Pri ukrytí ponuky, tak získavame väčšiu plochu, na ktorú je možné vypisovať požadované informácie.

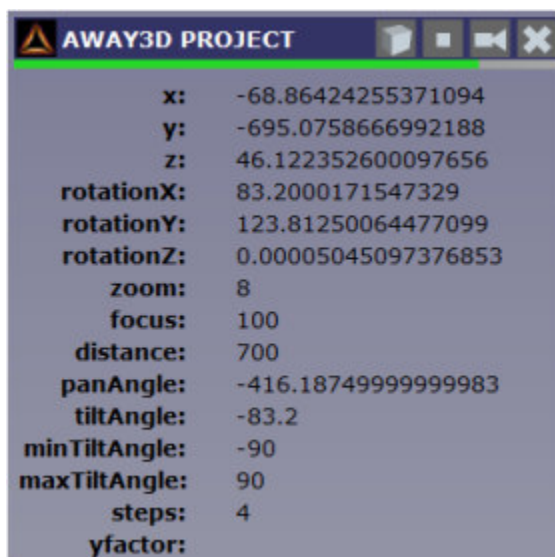
Aby bolo možné celú aplikáciu ovládať len za pomoci myši, bola doplnená špeciálna komponenta – vysúvacie menu (obrázok 7.2), ktoré za pomoci jednotlivých ikon - tlačidiel umožňuje plne nahradiť ovládanie klávesnicou, ale umožňuje aj pridávanie a ukrývanie jednotlivých častí modelu bez nutnosti otvárať ponuku hlavného menu, čím sa aplikácia stáva ešte intuitívnejšou.



Obrázok 7.2 – Vysúvne menu a jeho funkčné ikony

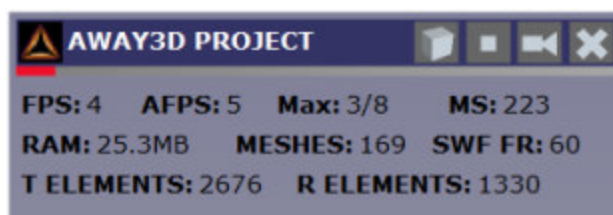
8 Testovanie aplikácie

Framework Away3D nám ponúka možnosť zobraziť štatistiky vytvoreného projektu, ak v okne aplikácie klikneme pravým tlačidlom myši na model FITkitu. Okno štatistík môžeme ľubovoľne presúvať po scéne, čo nám umožňuje sledovať rôzne sa meniace hodnoty, ale zároveň nebudeme ochudobnení ani o pozorovanie modelu. Máme tak možnosť podrobnejšie sledovať zmeny rotácie a pohybov na scéne, rôzne vzdialenosti a uhly kamery (obrázok 8.1), ďalším možným zobrazením štatistík je zoznam vykreslených útvarov s detailnejším popisom.



Obrázok 8.1 – Štatistiky aktuálnych parametrov kamery

Pre nás najzaujímavejšími štatistikami sú ale podrobnosti tretieho zobrazenia (obrázok 8.2), ktoré nás informuje o aktuálnom počte snímkov za sekundu (FPS), čo je jeden z najdôležitejších údajov, pretože výsledkom tohto parametra v praxi je plynulosť chodu celej aplikácie. Ďalším zaujímavým údajom je priemerný počet snímkov za sekundu (AFPS), ktorý vypočítava priemerné hodnoty počas interakcie s modelom. V závislosti na hodnote týchto veličín v porovnaní s počtom snímkov za sekundu v nami vygenerovanom SWF súbore (SWF FR) sa mení hodnota prúžku nad týmito údajmi, čo nám umožňuje sledovať reálny dopad na výkon aplikácie. Čím viac reálnych snímkov za sekundu sa podarí aplikácii dosiahnuť, tým bude hodnota tohto prúžku viac vpravo, teda zelená. Naopak, pri nižšom počte snímkov sa hodnota prúžku začne posúvať vľavo, jasne zelená farba začne tmavnúť, až môže dosiahnuť červené hodnoty (tento rozdiel môžeme vidieť na obrázkoch 8.1 a 8.2). Taktiež môžeme sledovať nároky na využitie operačnej pamäte RAM, ako aj počet objektov nachádzajúcich sa na scéne (MESHES).



Obrázok 8.2 – Štatistiky aktuálnych parametrov scény

8.1 Plynulosť scény v závislosti na počte vykresľovaných objektov

V poslednom riadku štatistík (obrázok 8.2) môžeme vidieť celkový počet vykresľovaných polygónov na scéne (*T ELEMENTS*) a počet naposledy (pri renderovaní posledného snímku) vykreslených polygónov (*R ELEMENTS*), v konečnom dôsledku zistíme, že práve tieto dva údaje v najväčšej miere ovplyvňujú plynulý chod celej aplikácie, pretože s počtom vykresľovaných polygónov súvisia aj nároky na výpočetný čas procesora, keďže Flash zatiaľ podporuje hardvérové urýchlenie vykresľovania len čiastočne. Udáva sa, že maximálny počet polygónov, pri ktorom je aplikácia ešte „použiteľná“ (plynulá) je 3000. Musíme totiž brať do úvahy, že pri zmene scény sú prepočítavané parametre všetkých polygónov, čo aj dvojjadrový procesor dokáže naozaj poriadne vyťažiť.

Odstránený prvok	<i>T ELEMENTS</i>	<i>R ELEMENTS</i>	<i>MESHES</i>	<i>AFPS</i>
Žiadny	2676	1200 - 1400	169	7
DIN 01-50	2064	1000 - 1100	118	10
+ DIN 01-40	1572	750 - 900	77	12
+ DIN 01-12	1416	650 - 750	64	14
+ LCD displej	1276	600 - 700	59	17
+ Klávesnica	968	400 - 500	40	23

Tabuľka 8.1 – Zhrnutie parametrov scény v závislosti na zobrazovaných komponentoch. (Testovacia zostava: Intel Core2 Duo T5870 (2.0GHz), 4GB RAM, Windows 7 Professional 64-bitový.)

Testy boli prevedené s postupne odoberanými objektmi zo scény, ktoré sú vymenované v prvom stĺpci tabuľky 8.1. K dosiahnutiu priemerných hodnôt bolo nevyhnutné so scénou pohybovať, preto bola ovládaná klávesnicou (tlačidlo „D“ – rotácia doprava) a otočená o 360 stupňov okolo osi y. Z tabuľky jasne vidíme ako výrazne sa mení počet polygónov na scéne (druhý stĺpec) najmä pri odstránení DIN konektorov, pretože tieto konektory sú tvorené najväčším počtom objektov (primitív), keď každá koncovka konektora predstavuje samostatný objekt. V treťom stĺpci zase môžeme sledovať, ako sa mení počet polygónov, ktoré je nutné pri každom snímku znova prepočítavať. Postupným odstraňovaním prvkov klesá počet objektov na scéne (štvrtý stĺpec), zároveň však stúpa priemerný počet snímkov za sekundu (posledný stĺpec), čím sa zvyšuje plynulosť aplikácie. Ďalším prvkom, ktorý scénu výrazne spomaľuje je klávesnica, ktorá aj keď nie je zložená z takého veľkého počtu objektov ako DIN konektory, každé z jej 16 tlačidiel má na svojom povrchu textúru. Takto sa nám postupne podarilo zvýšiť priemerný počet snímkov za sekundu na scéne z pôvodných 7, keď na scéne boli zobrazené všetky objekty, až na 23, keď boli voliteľné objekty so scény odstránené, a tento rozdiel sa prejavil viditeľne aj v praxi pri práci s aplikáciou. Zatiaľ čo pri zobrazení všetkých objektov je dojem z aplikácie taký, že reaguje trochu pomalšie (spomalene), čo je spôsobené zložitou a počtom výpočtov, ktoré musí procesor vzhľadom na počet polygónov vykonať, tak pri odstránení voliteľných objektov je aplikácia už plne plynulá.

8.2 Optimalizácia

Aby došlo aspoň k malému zrýchleniu aplikácie pri zobrazení veľkého počtu modelov na scéne, bolo aplikovaných niekoľko optimalizácií.

V prípade, že je zobrazená klávesnica FITkitu, objekty operačnej pamäte DRAM a čipu FPGA sú zo scény odstránené, pretože klávesnica ich zakrýva, takže v podstate takmer ani nie sú viditeľné. Je tu len zopár špecifických uhlov pohľadu, kedy by bolo možné tieto objekty vidieť, aj keď je na scéne zobrazená klávesnica, avšak textúra použitá na matičnej doske obsahuje náhradu týchto komponentov, takže zároveň vytvára dojem, že sa na scéne pod klávesnicou nachádzajú nepretržite. Pri odstránení komponentov klávesnice sú modely operačnej pamäte DRAM a čipu FPGA súčasne v tej istej chvíli do scény vložené. Rovnakým spôsobom sa ukrývajú čipy U20 a U21 pod LDC displejom. Týmto dosiahneme miernu optimalizáciu celkového počtu polygónov. Ako už ale z testov vyplýva, najvýraznejším spôsobom je scéna optimalizovaná pri odstránení veľkého počtu objektov, čiže odstránenie DIN konektorov zo scény je tou najlepšou optimalizáciou zobrazovanej scény.

Ďalším faktorom, ktorý ovplyvňuje celkovú náročnosť na výpočetný čas procesora sú textúry, ktoré môžeme vhodným stupňom kompresie, ako aj správne zvoleným formátom optimalizovať, čo sa týka veľkosti súboru. Ako najlepšia voľba sa ukázal stratový formát JPEG, ktorý aj napriek vyššej kompresii podáva stále dostatočujúcu kvalitu obrazu, ale zároveň výsledné súbory majú menšiu veľkosť ako súbory PNG alebo GIF. Dôležité je taktiež rozlíšenie textúry, preto bolo potrebné nájsť vhodný pomer medzi rozlíšením a výslednou kvalitou zobrazenej textúry na cieľovom objekte. V najvyššom rozlíšení je textúra prednej strany matičnej dosky (pre ukážku - veľkosť súboru JPEG: 98KB, GIF: 161KB, PNG: 398KB), ktorá obsahuje viacero komponentov, ktoré aplikácia fyzicky nezobrazuje, preto je v tomto prípade priam nevyhnutná dobrá kvalita výsledného obrazu, aby bolo možné rozlíšiť jednotlivé komponenty zobrazené na textúre.

8.3 Kompatibilita

Výsledná aplikácia bola drobnými úpravami prerobená z pôvodnej vyvíjanej s frameworkom Flex SDK 3.4 na 3.5, a nakoniec aj v súčasnosti najaktuálnejšej verzii Flex SDK 4. Prechod na najnovšiu verziu vyžadoval len zopár úprav zdrojového kódu, ale taktiež bolo nevyhnutné prejsť od kompilácie pre Flash Player 9, na taktiež poslednú radu Flash Player 10, pričom všetko funguje bezchybne aj pri kompilácii pre verziu až 10.1.

Aplikácia bola úspešne otestovaná v internetových prehliadačoch Mozilla Firefox 3.6 a 4, Internet Explorer 8 a 9, Opera 10 a 11, Google Chrome 9 a 10, Apple Safari 4 a 5. Keďže aplikácia je platformovo nezávislá, funguje na všetkých operačných systémoch, pre ktoré je dostupný niektorý z internetových prehliadačov, otestované na Windows XP, Windows Vista, Windows 7, Linux Mandriva, Linux (K)Ubuntu, MacOS X 10.5 (Leopard) a 10.6 (Snow Leopard).

9 Možnosti rozšírenia aplikácie

Framework Away3D nám vďaka tomu, že je vytvorený v programovacom jazyku ActionScript 3.0 umožňuje rozširovať nielen samotné súbory knižníc, príkladom čoho môže byť aj upravený vzhľad tlačidiel aplikácie, ktoré sú zmenené na základe úprav vykonaných v súbore „away3d/test/Button.as“, kde je definovaná trieda tlačidla „Button“, ale ponúka aj niekoľko možností, ako rozšíriť funkcionality aplikácie v budúcnosti.

Aplikácia by mohla byť rozšírená spôsobom, ktorý by umožňoval ovládanie FITkitu pripojeného k serveru – stisk tlačidla klávesnice by sa emuloval na FITkite, a naopak – displej by zobrazoval aktuálny stav. Používateľ by si tak mohol vyskúšať prácu s FITkitom prostredníctvom on-line rozhrania.

Zobrazovanie textu v 3D priestore je problematické, pretože hneď prvým nedostatkom, s ktorým sa Away3D potýka je nutnosť vloženia fontu z externého SWF súboru, z ktorého musí byť následne extrahovaný. Pri zobrazovaní textu v aplikácii sa nám ponúka niekoľko možností, pričom medzi dve, od ktorých sú odvodené aj ostatné patrí vytvorenie textového poľa pomocou knižníc frameworku Flex, ale tento text je ako pôvodný Flash schopný pracovať len v 2D prostredí, čím automaticky prekrýva všetok 3D obsah, keďže 3D vrstva je zobrazená až za (pod) 2D vrstvou. Druhým spôsobom, je vytvoriť text priamo v 3D prostredí Away3D, ale táto možnosť zasa vyžaduje už spomínaný import fontu do prostredia aplikácie, s ktorým už musí byť aplikácia aj skompilovaná. Výhodou tejto možnosti je najmä to, že s textom tak môžeme pracovať rovnako ako s ktorýmkoľvek iným 3D objektom.

10 Záver

Bakalárska práca si kládla za cieľ vytvoriť interaktívnu aplikáciu, ktorá by slúžila na prezentovanie platformy FITkit. Vďaka technológii Flash sa stala aplikácia realitou.

Pri tvorbe som sa zoznámil podrobnejšie s programovacím jazykom ActionScript, ktorý sa stal mojim cenným pomocníkom aj pri využívaní frameworku Away3D, s pomocou ktorého mi bolo umožnené využívať trojrozmerné objekty v inak dvojrozmernom prostredí Flashu. Objavil som tak nespočetné množstvo možností, ktoré táto technológia ponúka nie len, čo sa týka prezentácie interaktívneho obsahu na internete, ale o značné poznatky som sa obohatil aj v oblasti 3D grafiky.

Istý čas som „stratil“ pri tvorbe modelu v nástroji Blender, a pri vývoji aplikácie v pôvodne odporúčanom frameworku Flash Sandy, kde som veľmi rýchlo narazil na problémy s nesprávnym vykresľovaním polygónov, aj napriek odporúčaným optimalizáciám modelu sa niektoré polygóny nevykresľovali správne, tak som teda pristúpil k tvorbe modelu pomocou dostupných primitív, avšak tu som natrafil na ďalší problém tohto frameworku, ktorým bolo nesprávne prekrývanie sa objektov v závislosti na hĺbke scény (Z-sorting). Po týchto problémoch som bol nútený pátrať po inej možnosti, ktorou sa stal práve framework Away3D, ten ma už pri vytváraní prvej kostry modelu veľmi potešil, pretože ponúkal aj spôsob vykresľovania scény, ktorý je síce náročnejší na hardvér počítača, ale rieši problém z-sortingu. Ako som pri ďalšom vývoji aplikácie zistil, tak v konečnom dôsledku som len získal, pretože Away3D ponúka podstatne viac, ale aj modernejších možností.

Aplikácia ponúka niekoľko možností rozšírenia v budúcnosti, a to nie len vo forme jazykových mutácií pre zvýšenie prístupnosti, ale aj vďaka prepojeniu cez ethernetové rozhranie, čím by bolo umožnené demonštrovať a emulovať možnosti platformy FITkit on-line. Posledným atraktívnym rozšírením je práve možnosť zobrazovania textu na displeji. Po dokončení aplikácie som sa sám rozhodol si jednu z ponúkaných možností vyskúšať, čo sa mi aj podarilo implementovať, a tak som využil túto možnosť pri vytvorení dema, ktoré prezentuje platformu FITkit. Prišiel som tu ale na nedostatok frameworku, ktorý spočíva v prekrývaní písmen, text sa teda síce vykresľuje správne, ale pri rotácii modelu prekrýva všetky ostatné objekty. Čiastočným riešením by bolo naniesenie textu na displej vo forme textúry, ale keby sme napríklad chceli simulovať pohyb textu na displeji, tak by bola potrebná pre každý „pohyb“ nová textúra, čím by zároveň aplikácia bola obmedzená len na text, ktorý by bol obsahom týchto vopred vytvorených textúr.

Osobne ako najväčší prínos tejto práce považujem nielen prehĺbenie mojich vedomostí a skúseností s programovacím jazykom ActionScript, ale najmä to, že som objavil veľký potenciál 3D grafiky vo Flashi, ktorú môžem využiť aj v iných oblastiach (napr. tvorba internetových stránok).

Rovnako sa mi páči aj predstava, že aplikácia bude umiestnená na oficiálnych stránkach platformy FITkit, čím pomôže všetkým záujemcom o túto platformu získať reálny pohľad. Náhlady stránok, do ktorých je možné zakomponovať aplikáciu nájdeme ako prílohy A a B.

Literatúra

- [1] Kolektív autorů. Úvod - FITkit [online]. [cit. 2011-04-12]. Dostupné z WWW: <<http://merlin.fit.vutbr.cz/FITkit/>>.
- [2] LIPMAN, Robert. VRML Plugin and Browser Detector [online]. 29.03.2011 [cit. 2011-04-12]. Dostupné z WWW: <<http://cic.nist.gov/vrml/vbdetect.html>>.
- [3] VRML - Wikipedie [online]. 20.01.2011 [cit. 2011-04-12]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/VRML>>.
- [4] Acrobat 3D Developer Center [online]. [cit. 2011-04-12]. Dostupné z WWW: <<http://www.adobe.com/devnet/acrobat3d.html>>.
- [5] WHAT ID DOES | Turntool [online]. [cit. 2011-04-12]. Dostupné z WWW: <http://www.turntool.com/view_3d/what_it_does/>.
- [6] SHANKLAND, Stephen. Battle lines drawn for 3D on the Web [online]. 29.11.2010 [cit. 2011-04-12]. Dostupné z WWW: <http://news.cnet.com/8301-30685_3-20023350-264.html>.
- [7] HALFACREE, Gareth. WebGL brings 3D acceleration to browsers [online]. [cit. 2011-04-12]. Dostupné z WWW: <<http://www.bit-tech.net/news/bits/2009/08/05/webgl-brings-3d-acceleration-to-browsers/1>>.
- [8] ZINDROS, Dionysis. Why WebGL? [online]. 03.01.2011 [cit. 2011-04-12]. Dostupné z WWW: <<http://vl.kamibu.com/2011/01/why-webgl/>>.
- [9] Adobe Flash - Wikipedia, the free encyclopedia [online]. 04.04.2011 [cit. 2011-04-13]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Adobe_Flash>.
- [10] Comparison of HTML5 and Flash - Wikipedia, the free encyclopedia [online]. 31.03.2011 [cit. 2011-04-13]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Comparison_of_HTML5_and_Flash#Current_availability>.
- [11] Adobe Flash - Wikipedie [online]. 13.12.2010 [cit. 2011-04-13]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Adobe_Flash>.
- [12] Adobe – Flash Player: Súhrnná stránka [online]. 14.07.2009 [cit. 2011-04-13]. Dostupné z WWW: <<http://www.adobe.com/sk/products/flashplayer/>>.
- [13] Adobe – Nainstalovat aplikaci Adobe Flash Player [online]. 14.03.2009 [cit. 2011-04-13]. Dostupné z WWW: <<http://get.adobe.com/cz/flashplayer/>>.
- [14] ActionScript - Wikipedia, the free encyclopedia [online]. 31.03.2011 [cit. 2011-04-14]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/ActionScript>>.
- [15] Digital_rights_management – Wikipedie [online]. 09.01.2011 [cit. 2011-04-14]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Digital_rights_management>.
- [16] About – Flex SDK – Adobe Open Source [online]. 21.02.2008 [cit. 2011-04-14]. Dostupné z WWW: <<http://opensource.adobe.com/wiki/display/flexsdk/About>>.
- [17] Open source software for web developers [online]. 2011 [cit. 2011-04-14]. Dostupné z WWW: <<http://www.adobe.com/products/flex/overview/>>.
- [18] Download Flex 4 – Flex SDK – Adobe Open Source [online]. 01.07.2010 [cit. 2011-04-14]. Dostupné z WWW: <<http://opensource.adobe.com/wiki/display/flexsdk/Download+Flex+4>>.
- [19] Main Page – Flash Develop [online]. 2010 [cit. 2011-04-14]. Dostupné z WWW: <http://www.flashdevelop.org/wikidocs/index.php?title=Main_Page>.
- [20] Exploring 3D in Flash [online]. [cit. 2011-04-18]. Dostupné z WWW: <<http://www.kirupa.com/developer/actionscript/3dexplore.htm>>.

- [21] Away3D – Wikipedia, the free encyclopedia [online]. 28.02.2011 [cit. 2011-04-18].
Dostupné z WWW: <<http://en.wikipedia.org/wiki/Away3D>>.
- [22] Sandy 3D engine (AS2 & AS3) for Adobe Flash/about[online]. [cit. 2011-04-18].
Dostupné z WWW: <<http://www.flashesandy.org/about>>.
- [23] Away3D and Papervision3D Performance Test [online]. 01.05.2010 [cit. 2011-04-20].
Dostupné z WWW: <<http://inside.silverbase.net/away3d-papervision3d-performance/>>.
- [24] Coobico: 3D in Flash: Papervision, Away3D and Sandy3D [online]. [cit. 2011-04-20].
Dostupné z WWW:
<http://coobico.com/index.php/blog/article/3d_in_flash_papervision_away3d_and_sandy3d/en/>.
- [25] Adobe Forums: best flash 3D engine? [online]. 15.03.2011 [cit. 2011-04-20]. Dostupné z
WWW: <<http://forums.adobe.com/thread/645219>>.
- [26] Level of detail - Wikipedia, the free encyclopedia [online]. 13.02.2011 [cit. 2011-04-21].
Dostupné z WWW: <http://en.wikipedia.org/wiki/Level_of_detail>.
- [27] Z-order - Wikipedia, the free encyclopedia [online]. 16.01.2011 [cit. 2011-04-21].
Dostupné z WWW: <<http://en.wikipedia.org/wiki/Z-order>>.
- [28] Away3D Flash Engine >> Downloads [online]. [cit. 2011-04-21]. Dostupné z WWW:
<<http://away3d.com/downloads>>.
- [29] BRYNILDSEN, Jens C. Installing Away3D for Flash, Flex or Flashdevelop [online].
08.08.2008 [cit. 2011-04-22]. Dostupné z WWW:
<http://www.flashmagazine.com/Tutorials/detail/installing_away3d_for_flash_flex_or_flashdevelop/>.
- [30] BRYNILDSEN, Jens C. Away3D Basics 1 – A simple Away3D class explained [online].
20.08.2008 [cit. 2011-04-23]. Dostupné z WWW:
<http://www.flashmagazine.com/Tutorials/detail/away3d_basics_a_simple_away3d_class_explained/>.

Zoznam príloh

Príloha A - Náhľad podstránky „Hardware“ so zakomponovanou aplikáciou


Príloha B - Náhľad podstránky „Popis propojek“ so zakomponovanou aplikáciou

Príloha C - CD so zdrojovými kódmi a spustiteľnou aplikáciou

Príloha A

[Úvod](#) | [Hardware](#) | [Firmware](#) | [Návody](#) | [Aplikace](#) | [Tým](#) | [Diskuzní fórum](#) | [SVN](#)


Čeština / English

 **FITkit**

Login

Hardware

FITkit | Rozšiřující modul - Ethernet | Rozšiřující modul - RF 2.4GHz



Show/hide display

Show/hide keyboard

Show/hide DIN_01_12

Show/hide DIN_01_40

Show/hide DIN_01_50

Run demo

Controls:
Click and hold mouse
to rotate manually
W - look up
S - look down
A - look left
D - look right
Up - zoom in
Down - zoom out

Help

Kit obsahuje

- MCU rodiny MSP430 (Texas Instruments)
- FPGA Spartan 3 XC3S50-4PQ208C nebo XC3S400-4PQ208C (Xilinx)
- USB převodník FT232C
- audio rozhraní
- konektory PS2
- rozhraní VGA
- konektor RS232
- DRAM 8x8Mbit
- Klávesnice
- Řádkový LCD displej
- Rozšiřující konektory

Schémata

- FITkit verze 1.0
- FITkit verze 1.2
- FITkit verze 2.0
- Popis propojek

1. Poznámky

- **Nastavení kontrastu LCD**
V případě, že na LCD displeji nic nevidíte, je zapotřebí poladit kontrast
- **Programování a bootování FITkitu**
Dokument popisující způsob programování FITkitu (mikrokontroleru a FPGA) a jeho bootování.
- **FITkit verze 2.0**
Tento dokument popisuje odlišnosti FITkitu verze 2.x od starších kitů verze 1.x

[úvodní strana](#) | [mapa stránek](#) | [RSS](#)

Zobrazeno: 3845x | Naposledy: 25.10.2010 17:37:03

Dokumenty a zdrojové kódy jsou dostupné pod BSD licenci

© 2006 - 2009 Zdeněk Vašíček, Fakulta informačních technologií VUT Brno

Príloha B



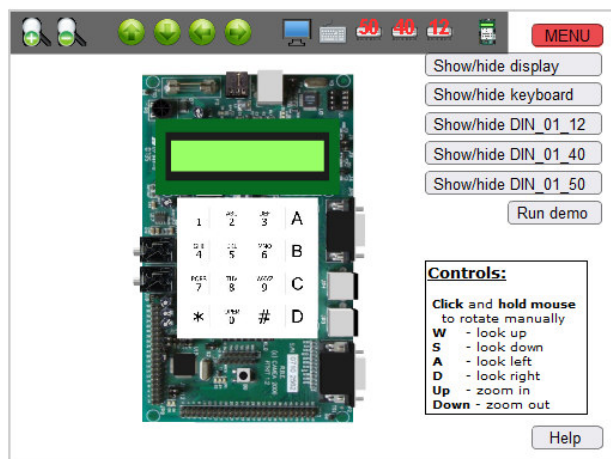
FITkit

Login

Hardware / Popis propojek

Autor: Karel Slaný (slany@fit.vutbr.cz)

Update: 18.3.2009



- Propojka J7 umožňuje deaktivovat USB převodník. Je-li propojka uzavřena, je FITkit odpojen od USB sběrnice (jako by byl odpojen kabel).
- Propojky J8, J9 tvoří pár. Pokud jsou vyjmuty, provede se reset MCU a program v MCU začne běžet bez nutnosti mít spuštěný terminál. Při programování musí být obě propojky uzavřeny.
- Propojky J11 a J12 slouží k připojení MCU k sériovému portu, který poskytuje USB převodník. Pokud jsou vyjmuty, nebude možné komunikovat s MCU pomocí terminálu.
- Propojky J13, J14, J15, J16 slouží k propojení MCU s audio zesilovači. Propojka J19 slouží jako MUTE v otevřeném stavu (propojka vyjmuta). Pokud je tato propojka uzavřena, zesilovač je aktivní.

Spoj	Popis
	Ukryt zvyrazneni
J2	připojení pravého kanálu JP6 na mikrofonní vstup audio kodeku (FITkit 2.0)
J3	připojení pravého kanálu JP6 na linkový vstup audio kodeku (FITkit 2.0)
J4	připojení levého kanálu JP6 na linkový vstup audio kodeku (FITkit 2.0)
J5	povolení přerušení z FPGA do MCU (FITkit 2.0)
J6	aktivace PS2, RS232, VGA
J7	propojení uvede USB převodník FT232RL do resetu
J8	povolení ovládání resetu MCU pomocí DTR
J9	povolení ovládání TCK MCU pomocí RTS
J11	RXD propojen na P3
J12	TXD propojen na P3
J13	propojení audio vstupu s portem MCU (AIR) (FITkit 1.x)
J14	propojení audio vstupu s portem MCU (AIL) (FITkit 1.x)
J15	propojení audio výstupu s portem MCU (AOR) (FITkit 1.x)
J16	propojení audio výstupu s portem MCU (AOL) (FITkit 1.x)
J17	ref. napětí pro analog převodník (FITkit 1.x)
J18	povolení EEPROM paměti USB převodníku (FITkit 2.0)
J19	audio zesilovač (SHD - MUTE) (FITkit 1.x)